

# Routing algorithm of minimizing maximum link congestion on grid networks

Jun Xu · Jianfeng Yang · Chengcheng Guo ·  
Yann-Hang Lee · Duo Lu

© Springer Science+Business Media New York 2014

**Abstract** As regular topology networks, grid networks are widely adopted in network deployment. Link congestion and routing path length are two critical factors that affect the delay and throughput of a network. In this paper, we study the routing problem in grid networks concerning these two factors. The main objective of our routing algorithms is to minimize the maximum link congestion. The shortest path minimum maximum (SPMM) link congestion and non-shortest path minimum maximum (NSPMM) link congestion routing problems are studied. The two problems are first formulated as integer optimization problems. Then, corresponding routing algorithms (SPMM and NSPMM routing algorithms) are proposed. For SPMM routing algorithm, the path length is optimal, while for NSPMM routing algorithm, the path is limited in a submesh. Thus, the path length can be bounded. At last, we compare the proposed routing algorithms under different scenarios with other popular routing algorithms (RowColumn, ZigZag, Greedy, Random routing algorithms). The performances are evaluated through different

metrics including link congestion, path length, path congestion, path congestion to path length ratio, delay and throughput.

**Keywords** Grid networks · Link congestion · Routing algorithm · Scheduling

## 1 Introduction

Network congestion occurs when some nodes or links carry too much traffic. It derogates network performances in several aspects, such as increasing the end-to-end delay, packet loss, and decreasing throughput. Reducing maximum link/node congestion and balancing network traffic are fundamental problems in network management. For example, one of the main objectives of Traffic Engineering (TE) [1] is load balancing. The load balancing problem has been widely studied and some researchers surveyed the load balancing problem on Internet [2], IP/MPLS [3], WiMAX [4], Wireless Sensor Networks [5, 6], P2P networks [7] and MANET [8, 9]. Related schemes on network congestion include rate control [10], queue management [11], topology control [12] and joint congestion control and medium access control [13]. Besides, designing a high performance routing algorithm is also an efficient method, such as multipath routing [14, 15]. However, the multipath routing method induces packet reordering problem [16].

In this paper, we design routing algorithms aiming to minimize the network congestion. The networks we study are the 2-dimensional grid topology networks. As studied in paper [17], in wireless mesh networks, grid topology networks are much more suitable for large scale mesh deployment compared to random topology networks. When formulating and analyzing wireless communication

---

J. Xu · J. Yang (✉) · C. Guo  
School of Electronic Information, Wuhan University, Wuhan,  
Hubei, China  
e-mail: yjf@whu.edu.cn

J. Xu  
e-mail: eisxujun@whu.edu.cn

C. Guo  
e-mail: netccg@whu.edu.cn

Y.-H. Lee · D. Lu  
Computer Science Department, Arizona State University,  
Tempe, AZ, USA  
e-mail: yhlee@asu.edu

D. Lu  
e-mail: duolv@asu.edu

problems, the 2-dimensional grid network provides a framework. In other communication networks, grid networks can be the testing topologies for routing and scheduling algorithms [18].

In the study [19], Leighton, Maggs, and Rao proved that there exist packet scheduling algorithms completing the transmissions of all the packets within duration  $O(c + d)$ , where  $c$  is the maximum link congestion of the network and  $d$  is the largest path length. Motivated by this result, many researchers tried to find effective routing algorithms aiming to minimize the maximum link congestion by adopting routing paths as short as possible.

Finding routing algorithms to minimize the maximum link congestion has been studied in game theory area as the bottleneck routing problem [18, 20–22]. The bottleneck node/link of a routing path is the most congested node/link on the path. In the routing games studied in [18, 20, 22], the objective of each player is to find a path with the minimum bottleneck link congestion. In [18], it allows the path to take some bends which implies that the routing paths adopted are not always the shortest paths. The price of anarchy (PoA) is proportional to the number of bends. In [22], the PoA is  $O(l + \log(n))$  where  $l$  is the length of the longest path in the player's strategy sets, and  $n$  is the size of the network. In [20], the splittable and unsplittable bottleneck games are studied. In [21], each user aims to minimize the sum of the maximum link congestion and service cost of the chosen path. In [23], the bottleneck congestion games are discussed, the aim is to minimize the maximum resource congestion of the network. In the routing game studied in [24], any player  $i$  tries to minimize its  $c_i + d_i$  value which represents the sum of the congestion and path length. Although the studies mentioned above tried to minimize the maximum link congestion of a network, all of them considered only the maximum link congestion of its routing path, while ignored the path congestion (which is the summation of all the link congestions along the path). However, this may result in a higher maximum link congestion of the network. An example is shown in Fig. 1. The numbers on the links are the current link congestions. Let flow  $f_i$  make a

decision before flow  $f_j$ . If each flow only considers the bottleneck link congestion of the chosen path, then for flow  $f_i$ , the bottleneck link congestion of all the routing paths are 2. It randomly chooses the path  $p_i = \{s_i, v_3, v_2, d_i\}$ . Now the congestions on links  $(s_i, v_3)$ ,  $(v_3, v_2)$ , and  $(v_2, d_i)$  are increased to 3. The maximum link congestion of the network is 3. Now, consider flow  $f_j$ , no matter which path it chooses, it will increase the maximum link congestion of the network from 3 to 4. However, if the path congestion is considered after the bottleneck link congestion, then flow  $f_i$  should choose the path  $p_i = \{s_i, v_1, v_2, d_i\}$  which has the minimum path congestion. Now, for flow  $f_j$ , it will choose path  $p_j = \{s_j, s_i, v_3, d_j\}$  which has the minimum bottleneck path congestion. In this situation, the maximum link congestion of the network is only 3, which is smaller than 4.

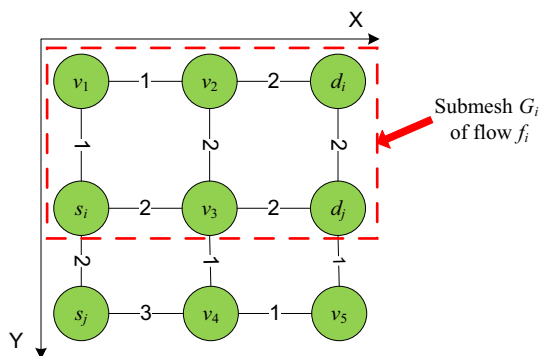
Motivated by the above observation, we consider not only the maximum link congestion but also the path congestion when selecting path for a flow in a grid network to minimize the maximum link congestion of the network.

In this paper, we propose two routing algorithms which are denoted by SPMM and NSPMM routing algorithms, corresponding to the two routing problems.

The evaluations of the proposed routing algorithms are executed under three different traffic patterns: Partial Permutation Traffic Pattern, General Traffic Pattern and the Specific Traffic Pattern. For the Partial Permutation Traffic Pattern, no two flows in  $F$  share the same source or destination node, that is, for any two flows  $f_i$  and  $f_j$ , we have that  $s_i \neq s_j$ ,  $s_i \neq d_j$ ,  $d_i \neq s_j$  and  $d_i \neq d_j$ . For the General Traffic Pattern, the source and destination nodes for any flow can be randomly selected. And more than one flow can share the same source or destination node. The construction of the Specific Traffic Pattern will be introduced in Sect. 5.

Our contributions include:

- The mathematical formulations of the SPMM and NSPMM problems. In both of the problems, some constraints such as the flow conservation constraints should be met.
- The SPMM and NSPMM routing algorithms. The SPMM routing algorithm tries to find the shortest path. However by NSPMM routing algorithm, the paths are not always the shortest ones. The aim of the two routing algorithms is the same.
- Performance evaluation. The SPMM and NSPMM routing algorithms are compared with other popular routing algorithms designed for grid networks, namely the RowColumn, ZigZag, Random and Greedy routing algorithms. The performance metrics we evaluate include link congestion, path length, path congestion, path congestion to path length ratio, delay and throughput.



**Fig. 1** An example of grid networks

The paper is organized as follows. In Sect. 2, we introduce some related works of the routing algorithms on grid networks. In Sect. 3, we present the network model and formalize the SPMM and NSPMM routing problems. The routing algorithms are described in Sect. 4. In Sect. 5, we evaluate the performances of the SPMM and NSPMM routing algorithms by comparing them with some existing routing algorithms designed for grid networks. At last, in Sect. 6, we conclude this paper.

## 2 Related work

Routing algorithms have been studied widely in different kind of networks, such as the disruption tolerant networks [25], delay tolerant networks [26], cognitive radio networks [27], wireless vehicular sensor networks [28], data centers [29] and multicast routing protocol [30]. Routing is an essential management mechanism for supporting Qos [31].

For grid networks, some routing algorithms with different objectives have been studied [18, 32–36].

The authors in [18] have studied the routing games in grid networks, each player aims to minimize the maximum link congestion of its routing path. The ultimate objective is to minimize the maximum link congestion of the network which is the same as that in this paper.

In [32], to minimize the makespan of a set of packets, the authors proposed the RowColumn routing algorithm combining the longest path first scheduling algorithm for the partial permutation traffic pattern. The optimal makespan can be achieved when all the packets are starting to transmit simultaneously. However, in applications like industry control, each flow generates packets with a specified period, thus it's hard to guarantee that the packets are started to transmit simultaneously unless the flows are simple periodic [37]. Adopting the RowColumn routing algorithm, the maximum link congestion can be as high as  $\max\{n_1, n_2\}$  for the Partial Permutation Traffic Pattern. For the General Traffic Pattern, the maximum link congestion can be as high as  $n_{\max}$ , where  $n_{\max}$  represents the maximum times of the nodes in a row or column being source or destination nodes among all the rows and columns.

In [33] and [35], the objective is to maximize the reliability of packet transmissions with the assumption that all the links in a grid network have the same packet delivery probability. As shown in [33], the ZigZag shortest path routing algorithm is optimal in 2-dimensional grid networks. However, it is not optimal in the torus [34]. In [35], the MSP (Maximum-Shortest-Path) routing algorithm has been proposed to achieve the same goal. By MSP, a packet is always forwarded to the neighbor with the maximum number of shortest paths to the destination. Unlike the

ZigZag routing algorithm, MSP is optimal not only in 2-D, 3-D meshes, but also in  $N \times N$  torus with  $N$  being an even number greater than 4.

A local-information-based routing algorithm denoted by ZigZag has been presented in [36]. By this routing algorithm, any given path between two nodes can be transformed into a shortest path. This strategy is useful in scenarios where nodes can move around.

Another type of routing algorithms is Random routing strategy [33, 38]. By the random routing algorithm discussed in [33], a packet is forwarded randomly to a neighbor which is nearer to the destination node. In [38], a packet is forwarded randomly to a neighbor which is not necessarily nearer to the destination node. With the purpose of transmitting all the packets to their destinations as quickly as possible, the author in [39] has studied the random algorithms for packet routing on grid networks. These algorithms consider packet routing as well as packet scheduling. In paper [40], the authors have tried to control  $C$  (the maximum link congestion) and  $D$  (path length, which is called dilation in the paper) simultaneously. The network congestion (maximum link congestion) achieved by the algorithm in this paper is  $O(dC^* \log(n))$  and the stretch (ratio of path length to the shortest path length) is  $O(d^2)$ , where  $C^*$  is the optimal congestion and  $d$  is the dimension of the grid (mesh as used in the paper). Randomization is used in the algorithm.

Compared to the above studies, our work aims to minimize the maximum link congestion of the network. One of our routing algorithms achieves the optimal path length, in other words, the stretch (the ratio of the path length to the shortest path length [40]) of the selected path of a flow is 1 and it also keeps the maximum link congestion low. The other routing algorithm has a higher stretch. However, in some instances, it can keep the maximum link congestion lower than the routing algorithms with the optimal stretch.

## 3 Network model and problem formulation

### 3.1 Network model

A grid network is represented by an undirected graph  $G = (V, E)$ , where  $V$  is the set of nodes and  $E$  is the set of edges. For any node  $v \in V$ , its coordinate is represented by  $(v^x, v^y)$ . By convention, the grid networks studied are 2-dimensional topology networks.

For a  $n_1 \times n_2$  grid network, it means that the numbers of nodes in any row and any column are  $n_1$  and  $n_2$ , respectively. The flows in the network are represented by  $F = \{f_1, \dots, f_n\}$ . For any flow  $f_i \in F$ , it is represented by a pair of source and destination nodes  $\{s_i, d_i\}$  where  $s_i \neq d_i$ .

For any grid network, the origin point [with coordinates  $(0, 0)$ ] is in the left-up corner. The X-axis is the horizontal line while the Y-axis is the vertical line. The X-coordinate of a node is the horizontal distance (hops) from the node to the origin point. Similarly, the Y-coordinate of a node is the vertical distance (hops) to the origin point. Thus, for a flow  $f_i$  whose source node is  $s_i$  with coordinate  $(s_i^x, s_i^y)$  and destination node is  $d_i$  with coordinate  $(d_i^x, d_i^y)$ , the horizontal distance between the two nodes is  $h_{i,x} = |d_i^x - s_i^x|$  and the vertical distance is  $h_{i,y} = |d_i^y - s_i^y|$ . In Fig. 1, we show an example of grid networks. The coordinates of node  $v_1$  is  $(0,0)$ . Thus, pick a node, say  $v_3$ , its coordinates are  $(1,1)$ . The horizontal distance from  $v_3$  to  $v_1$  is 1, so is the vertical distance.

Considering that by sacrificing path length, a routing algorithm may achieve a lower maximum link congestion compared to routing algorithms adopting shorter paths, thus we study two routing problems, which are shortest path minimum maximum link congestion (SPMM) and non-shortest path minimum maximum link congestion (NSPMM) routing problems.

In the SPMM routing problem, only the shortest paths will be adopted. However, in the NSPMM problem, the routing path of any flow  $f_i$  is restricted within its submesh  $G_i$ . The submesh  $G_i$  is defined in the following.

**Definition 1** (*The submesh  $G_i$* ) For the flow  $f_i$  with source node  $s_i$  and destination node  $d_i$ , the submesh  $G_i$  includes all the nodes and links in the rectangle with  $s_i$  and  $d_i$  as two opposite diagonal nodes.

In Fig. 1, the submesh of  $f_i$  is the one in the red (dashed) rectangle, it includes nodes  $V_i = \{v_1, v_2, v_3, s_i, d_i, d_j\}$  and links  $E_i = \{(v_1, v_2), (v_2, d_i), (v_1, s_i), (v_2, v_3), (d_i, d_j), (s_i, v_3), (v_3, d_j)\}$ . Path  $p = \{s_i, v_1, v_2, v_3, d_j, d_i\}$  is a path restricted within the submesh  $G_i$ , while path  $p' = \{s_i, s_j, v_4, v_3, v_2, d_i\}$  is a path beyond the submesh  $G_i$ .

The formal definitions of the two routing problems are given as follows.

**Definition 2** (*SPMM Routing Problem*) Given a grid network  $G = (V, E)$  and a set of flows  $F = \{f_1, \dots, f_n\}$ . For each flow  $f_i \in F$ , find a shortest path  $p_i$  for it such that the maximum link congestion of the network is minimized.

**Definition 3** (*NSPMM Routing Problem*) Given a grid network  $G = (V, E)$  and a set of flows  $F = \{f_1, \dots, f_n\}$ . For each flow  $f_i \in F$ , in the submesh  $G_i$ , find a path  $p_i$  for it such that the maximum link congestion of the network is minimized.

Now, we give the following definitions which will be used frequently when describing the routing algorithms.

**Definition 4** (*Shortest Path*) For any flow  $f_i$ , a shortest path  $p_i$  between its source node  $s_i$  and destination node  $d_i$  includes exactly  $|s_i^x - d_i^x| + |s_i^y - d_i^y|$  links.

**Definition 5** (*Link Congestion*) The congestion  $l(e)$  on any bidirectional link  $e$  is the total number of paths including link  $e$ , that is,  $l(e) = \sum_{f_i \in F} r_{i,e}$  where  $r_{i,e}$  is 1 when  $p_i$  includes link  $e$ , otherwise, it is 0.

**Definition 6** (*Bottleneck Path Congestion*) The bottleneck path congestion  $b(p)$  of path  $p$  is defined as the maximum link congestion among all the links on  $p$ , that is,  $b(p) = \max_{e \in p} l(e)$ .

**Definition 7** (*Path Congestion*) The path congestion of path  $p$  is the summation of congestions of all the links included by  $p$ , that is,  $w(p) = \sum_{e \in p} l(e)$ .

## 3.2 Problem formulation

### 3.2.1 The SPMM routing problem

The minimizing maximum link congestion routing problem can be formulated as an integer optimization programming. In the problem formulation, the undirected graph can be viewed as a directed graph by transforming each edge into two directed edges, let  $E'$  represent these edges.

Note that, for  $(u, v) \in E'$  and  $(v, u) \in E'$ , they are the same edge in  $E$ , thus the congestion of a link in  $E$  is the summation of congestions on two links in  $E'$ .

Now let  $\Gamma_v^+$  and  $\Gamma_v^-$  be the incoming and outgoing links adjacent to node  $v \in V$ , respectively. Let  $e'$  be any directed link in  $E'$ . The problem then can be formulated as follows and the variables we need to obtain are  $r_{i,e'} (i = 1, \dots, n; e' \in E')$ .

$$\text{Min. Max. } \sum_{(u,v) \in E'} r_{i,(u,v)} + \sum_{(v,u) \in E'} r_{i,(v,u)} \quad (1)$$

Subject to:

$$\sum_{e' \in \Gamma_v^+} r_{i,e'} - \sum_{e' \in \Gamma_v^-} r_{i,e'} = \begin{cases} 1 & \text{if } v = s_i \\ -1 & \text{if } v = d_i; f_i \in F \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\sum_{e' \in p_i} r_{i,e'} = h_i^x + h_i^y; f_i \in F \quad (3)$$

$$r_{i,e'} \in \{0, 1\}, \quad i = 1, \dots, n; e' \in E' \quad (4)$$

Constraints (2) ensure flow conservation for each flow on each node. Constraints (3) ensure that each flow can

only adopt the shortest path. Constraints (4) ensure that each variable is a 0–1 integer. The above integer optimization programming problem is a variation of the multi-commodity flow problem. For the original multi-commodity flow problem, price-directed decomposition, resource-directed decomposition and partition methods are introduced in [40]. We refer the interested readers to [41] for a much more detailed discussion.

### 3.2.2 The NSPMM routing problem

The motivation to consider the NSPMM routing problem is that, with the sacrificing of the path length, we may achieve smaller maximum link congestion. An example is shown in Fig. 2. In Fig. 2(a), the path of flow  $f_3$  is not a shortest path and the network congestion is 1 after all the three flows choosing their paths. While in Fig. 2(b), the path is a shortest path, the network congestion is 2. The tradeoff for a shortest path is the increasing of the network congestion.

The NSPMM routing problem is formalized as follows.

$$\text{Min. Max. } \sum_{(u,v) \in E'} r_{i,(u,v)} + \sum_{(v,u) \in E'} r_{i,(v,u)} \quad (5)$$

Subject to:

$$\sum_{e' \in \Gamma_v^+} r_{i,e'} - \sum_{e' \in \Gamma_v^-} r_{i,e'} = \begin{cases} 1 & \text{if } v = s \\ -1 & \text{if } v = d_i; f_i \in F \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$v \in G_i; \text{ for } v \in p_i \quad (7)$$

$$r_{i,e'} \in \{0, 1\}, \quad i = 1, \dots, n; e' \in E' \quad (8)$$

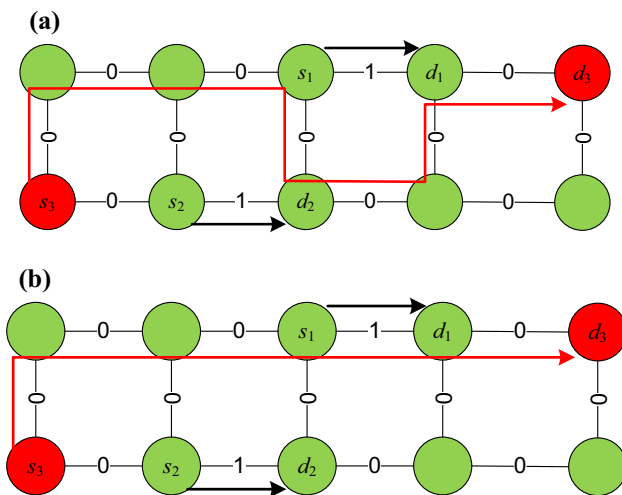


Fig. 2 NSPMM routing versus SPMM routing

Compared to the SPMM routing problem, instead of choosing a shortest path for each flow, a path restricted within the submesh is considered in the NSPMM routing problem. It is ensured by constraints (7) where each node  $v$  included by the path  $p_i$  is in the submesh  $G_i$ .

## 4 Routing algorithms

### 4.1 Routing algorithm: SPMM

In this section, we design a shortest path routing algorithm (SPMM) for grid networks. The ultimate goal is to minimize the maximum link congestion of the network.

Let  $P_i$  denote the set of shortest paths of flow  $f_i$ . The SPMM routing algorithm will select a path  $p_i \in P_i$  for it. When selecting the path, the following two conditions should be satisfied:

- The bottleneck path congestion of  $p_i$  is the minimum among all the routing paths in  $P_i$ , that is,  $b(p_i) = \min\{b(p) | p \in P_i\}$ .
- Let  $P'_i$  represent the set of paths in  $P_i$  whose bottleneck path congestions are equal to  $b(p_i)$ , that is  $P'_i = \{p | b(p) = b(p_i), p \in P_i\}$ . The path congestion of  $p_i$  is less than or equal to the minimum path congestion among all the paths in  $P'_i$ , that is,  $w(p_i) \leq \min\{w(p) | p \in P'_i\}$ .

The SPMM routing algorithm is a dynamic programming algorithm. Before discussing it in details, we need to introduce the concept of effective parent nodes and effective child nodes.

**Definition 8 (Effective Parent Nodes)** When considering the routing for flow  $f_i$ , the set of effective parent nodes  $P_v^i$  of node  $v$  are the nodes which locate in the submesh  $G_i$  and are nearer to the source node  $s_i$ . The following conditions are satisfied.

$$|u^x - s_i^x| + |u^y - s_i^y| < |v^x - s_i^x| + |v^y - s_i^y|, u \in P_v^i \text{ and } u \in G_i.$$

**Definition 9 (Effective Child Nodes)** When considering the routing for flow  $f_i$ , the effective child nodes  $C_v^i$  of node  $v$  are the nodes which locate in the submesh  $G_i$  are nearer to its destination node  $d_i$ . The following conditions are satisfied:

$$|u^x - d_i^x| + |u^y - d_i^y| < |v^x - d_i^x| + |v^y - d_i^y|, u \in C_v^i \text{ and } u \in G_i.$$

The pseudocode of the SPMM routing algorithm is shown in Algorithm 1. For different flows, the number of



shortest paths may be different. The number of shortest paths of flow  $f_i$  is  $\binom{h_{i,x} + h_{i,y}}{h_{i,x}}$ . For any flow, the more the choices are, the more likely it can avoid the congested links. It implies that the sequence of flows selecting paths is important to minimize the maximum link congestion. In algorithm 1,  $I(u)$  represents the id of node  $u$ . Each node in the network has a specified id. We can label the nodes row by row with increasing ids.

In line 1, we first sort the flows with the increasing number of shortest paths.

For each flow  $f_i \in F$ , we need to do the process from line 3 to line 28. Since a shortest path of flow  $f_i$  is within its submesh  $G_i$ , in line 3, we first get the submesh. The set  $S$  records the current visited nodes. The set  $T$  includes the nodes to be visited in the next round. It records the effective child nodes of the nodes in set  $S$ . The set  $Y$  records the unvisited nodes in  $V_i$ . Let  $b(p_{s_i,v})$  represent the bottleneck congestion of the selected path from node  $s_i$  to node  $v$ ,  $w(p_{s_i,v})$  represent the path congestion of the selected path from node  $s_i$  to node  $v$ . In fact, path  $p_{s_i,v}$  can be viewed as a subpath of the final selected path  $p_i$ .

From line 4 to line 7, we first initialize all the sets, and the values of  $b(p_{s_i,s_i})$  and  $w(p_{s_i,s_i})$ . Considering that in the submesh  $G_i$ , the destination node  $d_i$  is the farthest node from node  $s_i$ , so it's the last node to be visited. Thus, if set  $Y$  is not empty (which means node  $d_i$  hasn't been visited), we will keep doing the process from line 9 to line 19. In this process, for each node in set  $T$ , from line 10 to line 14, a precedent node  $pre(v)$  is selected for it from its effective parent nodes. First, in line 10, we get the effective parent nodes for node  $v$ . In line 11, we set the bottleneck path congestion for the subpath from source node  $s_i$  to node  $v$ ,  $P'_{v,i}$  records the parent nodes from which node  $v$  can obtain a path to  $s_i$  with the minimum bottleneck path congestion. In line 12, the algorithm updates the path congestion as the minimum path congestion of the subpath to  $s_i$  and selecting from  $P'_{v,i}$  the nodes from which node  $v$  can obtain a path whose path congestion is the minimum,  $P''_{v,i}$  records these parent nodes. If there're more than one nodes in set  $P''_{v,i}$ , select the one with the smaller node id as the precedent of node  $v$ , as shown in line 13 and line 14. From line 16 to line 18, the algorithm updates the corresponding sets. From line 20 to line 25, we can simply construct the path from the source node  $s_i$  to the destination node  $d_i$  by tracking the precedents from node  $d_i$  until we reach the source node. From line 26 to line 28, we increase the congestion of the links in path  $p_i$  by one.

---

**Algorithm 1:** SPM Routing Algorithm

---

**Input:** network topology  $G(V, E)$ ;  
Source and destination pairs:  $F = \{(s_1, d_1), \dots, (s_n, d_n)\}$ ;  
**Output:** routing path  $p_i$  ( $i = 1, 2, \dots, n$ )

- 1 Sort the flows in the increasing order of  $|P_i|$ ;
- 2 **for**  $i = 1$  to  $n$  **do**
- 3   Get the submesh  $G_i = (V_i, E_i)$ ;
- 4    $S \leftarrow null$ ;
- 5    $b(p_{s_i,s_i}) = 0$  and  $w(p_{s_i,s_i}) = 0$ ;
- 6    $T \leftarrow s_i$ ;
- 7    $Y \leftarrow V_i - T$ ;
- 8   **while**  $Y$  is not empty **do**
- 9     **for**  $v \in T$  **do**
- 10        $P'_{v,i} \leftarrow$  efficient parent nodes of node  $v$ ;
- 11        $b(p_{s_i,v}) \leftarrow \min_{u \in P'_{v,i}} \{\max\{b(p_{s_i,u}), l(u,v)\}\}$  ,
- $P'_{v,i} \leftarrow \operatorname{argmin}_{u \in P'_{v,i}} \{\max\{b(p_{s_i,u}), l(u,v)\}\}$  ;
- 12        $w(p_{s_i,v}) \leftarrow \min_{u \in P'_{v,i}} \{w(p_{s_i,u}) + l(u,v)\}$  ,
- $P''_{v,i} \leftarrow \operatorname{argmin}_{u \in P'_{v,i}} \{w(p_{s_i,u}) + l(u,v)\}$  ;
- 13        $\hat{p}_{v,i} \leftarrow \operatorname{argmin}_{u \in P''_{v,i}} \{I(u)\}$  ;
- 14        $pre(v) \leftarrow \hat{p}_{v,i}$  ;
- 15     **end**
- 16      $Y \leftarrow Y - T$ ;
- 17      $S \leftarrow T$ ;
- 18      $T \leftarrow$  set of the child nodes of node in  $S$ ;
- 19   **end**
- 20    $u \leftarrow d_i$  and  $p_i \leftarrow null$ ;
- 21   **while**  $u \neq s_i$  **do**
- 22      $p_i \leftarrow p_i \cup (pre(v), u)$ ;
- 23      $u \leftarrow pre(u)$
- 24   **end**
- 25    $p_i \leftarrow p_i \cup (pre(v), u)$
- 26   **for** each link  $e$  in  $p_i$  **do**
- 27      $l(e) \leftarrow l(e) + 1$ ;
- 28   **end**
- 29 **end**

---

#### 4.2 Routing algorithm: NSPMM

In this section, we present the NSPMM routing algorithm, the basic idea is the same as that of the SPM routing algorithm. The biggest difference between the two routing algorithms is that the path for a flow found by NSPMM routing algorithm is not always the shortest one. In order to control the path length, the path found by NSPMM routing algorithm is restricted within the submesh of each flow. The main process of the NSPMM routing algorithm is described as follows.

- We first sort the flows according to the areas of the submeshes. The area  $A_i$  of the submesh of flow  $f_i$  is calculated by  $A_i = (d_i^x - s_i^x) \times (d_i^y - s_i^y)$ .

- Then we find the paths with the minimum bottleneck path congestion. For flow  $f_i$ , the paths are a subset of  $\hat{P}_i$ , which are represented by  $\hat{P}'_i$ .
- Finally we find the path in  $\hat{P}'_i$  with the minimum total path congestion, which is the one we need.

Unlike the SPMM routing algorithm, it's hard to find the efficient parent nodes and efficient child nodes in the NSPMM routing problem. Thus, it's not practical to design a dynamic programming routing algorithm. However, by introducing the concept of critical congestion for each flow, we can design an efficient NSPMM routing algorithm. The definition of the critical congestion is given as follows.

**Definition 10** (*Critical Congestion*) For flow  $f_i \in F$ , at the time for it to find a routing path, for each link  $e \in E$ , the link congestion is  $l(e)$ . Let  $l_i^*$  be a link congestion value. Removing all the links with link congestions greater than  $l_i^*$ , there exists at least one path within the submesh  $G_i$  connecting node  $s_i$  and  $d_i$ . However, removing all the links with link congestion equal to or greater than  $l_i^*$ , we can't find any path within the submesh  $G_i$  connecting node  $s_i$  and  $d_i$ .

The pseudocode of the NSPMM routing algorithm is shown in Algorithm 2.

First, in line 1, all flows are sorted. For each flow  $f_i \in F$ , we need to execute the process from line 3 to line 11. For flow  $f_i$ , the algorithm first get its submesh  $G_i$  (line 3), then find its critical congestion in  $G_i$  (line 4). From line 5 to line 9, we remove the links in  $G_i$  with link congestion greater than the critical congestion. Now, in the residual network, all the links have link congestion equal to or less than the critical congestion. In line 10, using Dijkstra's algorithm [42], we can easily get the path with the minimum total path congestion. Note that, when running the Dijkstra's algorithm, we always pick the node with the smaller node id when there are more than one nodes that have the minimum path congestions in the paths from itself to the source node. In line 11, the congestions of the links in path  $p_i$  are increased by one.

---

**Algorithm 2:** NSPMM routing algorithm

---

**Input:** network topology  $G(V, E)$ ;  
Source and destination pairs:  $F = \{(s_1, d_1), \dots, (s_n, d_n)\}$ ;  
**Output:** routing path  $p_i$  ( $i = 1, 2, \dots, n$ )

```

1  Sort the flows in the increasing order of the areas of the submeshes;
2  for  $i=1$  to  $n$  do
3    Get the submesh  $G_i = (V_i, E_i)$ ;
4     $l_i \leftarrow \text{CriticalCongestion}(G_i)$ ;
5    for  $e \in E$  do
6      If  $l(e) > l_i$  do
7         $E_i \leftarrow E_i \setminus e$ ;
8    end
9  end
10 In the subgraph, use Dijkstra's algorithm to find
    path  $p_i$  whose total path congestion is minimum;
11  $l(e) \leftarrow l(e) + 1$  for each link  $e$  in  $p_i$ ;
12 end
```

---

Now, the only problem we need to solve is how to get the critical congestion for each flow. The authors in [43] proposed the method solving the problem of finding the maximum minimum bandwidth path. By slightly modifying the algorithm, we can use it to find the minimum maximum link congestion of a submesh. This link congestion will be the critical congestion of the corresponding flow.

---

**Algorithm 3:** CriticalCongestion( $G_i$ )

---

**Input:** subtopology  $G_i(V_i, E_i)$ ;  
Source and destination pair:  $(s_i, d_i)$ ;  
Link congestion  $l(e)$ ,  $e \in E_i$ ;  
**Output:** critical congestion  $l_i$

```

1  while  $\text{Iteration} < \lceil \log(|E_i|) \rceil$  do
2     $M \leftarrow \text{median}\{l(e)\} \text{ for } e \in E_i$ ;
3    Remove all the edges  $e$  with  $l(e) > M$ ;
4    If the graph is not  $s_i$ - $d_i$  connected then
5      Let  $V_1, \dots, V_q$  be the connected components;
6      Reinsert all the edges removed in this
        iteration;
7      Shrink  $V_1, \dots, V_q$ ;
8    end
9     $\text{Iteration} \leftarrow \text{Iteration} + 1$ ;
10 end
```

---

In order to make this paper self-contained and consider that our objective is different from [43], we give the pseudocode of how to get the critical congestion for a flow, as shown in Algorithm 3. For the detailed description of the algorithm, we refer the readers to paper [43]. As proved in the paper, the time complexity of algorithm CriticalCongestion( $G_i$ ) is  $O(|E_i|)$ .

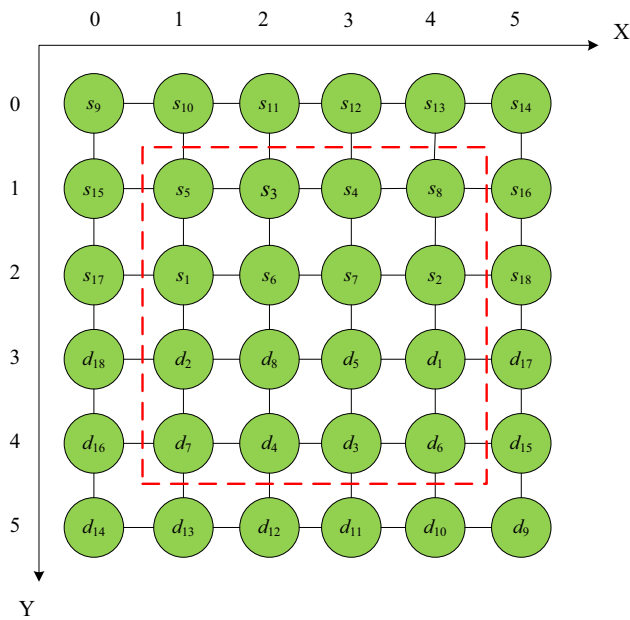
#### 4.3 Algorithm performance

**Theorem 1** For a  $n_1 \times n_2$  2-dimensional grid network with  $n$  flows, the time complexity of Algorithm 1 is  $O(nm_1n_2)$ .

*Proof* Sorting flows will take  $O(n \log(n))$  steps. In Algorithm 1, for any flow  $f_i$ , getting the efficient parent nodes of any node  $v$  can be accomplished with time complexity  $O(1)$  (using the coordinates of nodes). From line 8 to line 19, for flow  $f_i$ , each node in the submesh is visited exactly once, thus the time complexity is  $O(|V_i|)$ . The constructing of the routing path (line 20 to line 25) takes  $O(h_i)$  steps, where  $h_i$

**Table 1** initial configuration

Network size	Number of nodes	Number of links	Original number of flows	Initial link congestion
10 × 10	100	180	50	0
20 × 20	400	760	200	0

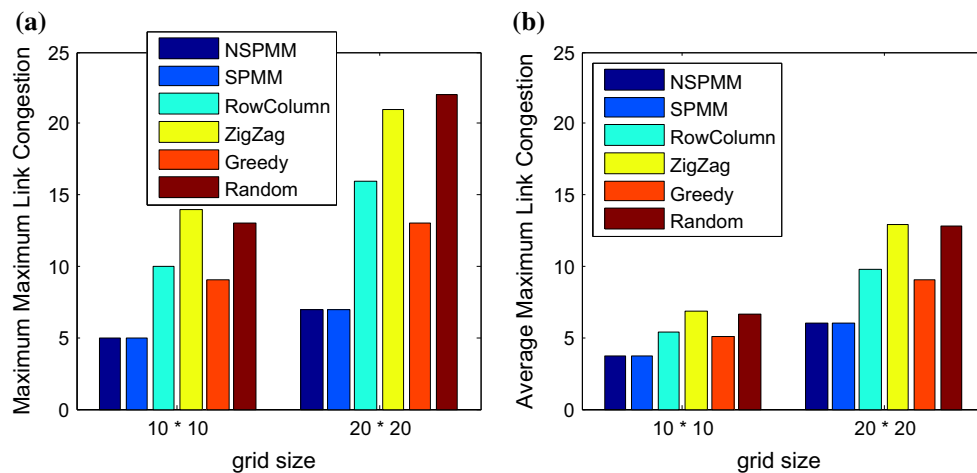


**Fig. 3** An example of flow construction with network size  $6 \times 6$

is the shortest path length for flow  $f_i$ . From line 26 to line 28, we update the congestions of links in path  $p_i$ , the time complexity is  $O(|E_i|)$ . The time complexity for flow  $f_i$  to get its routing path is  $O(|V_i| + h_i + |E_i|)$ . For a  $n_1 \times n_2$  grid, for any flow  $f_i$ , we have that,  $|V_i| \leq \frac{n_1 \times n_2}{2}$ ,  $h_i \leq \frac{n_1 \times n_2}{2}$ , and  $|E_i| \leq (n_1 - 1)n_2 + n_1(n_2 - 1)$ . Thus, we have that  $|V_i| + h_i + |E_i| \leq O(n_1 n_2)$ . In total, there are  $n$  flows, thus the time complexity of the SPMM routing algorithm is  $O(n n_1 n_2)$ .  $\square$

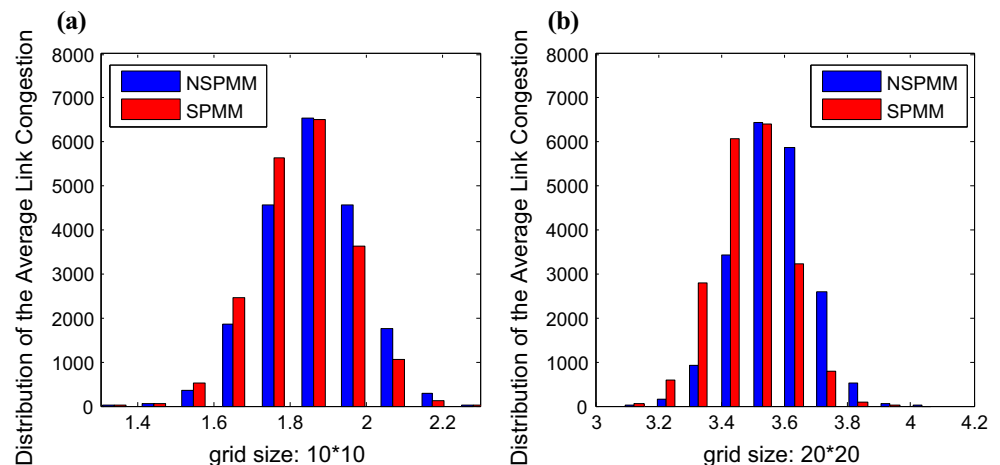
**Theorem 2** For a  $n_1 \times n_2$  2-dimensional grid network, the time complexity of Algorithm 2 is  $O(n \log(n) + m n_1 n_2 \log(n_1 n_2))$  where  $n$  is the total number of flows.

*Proof* First, sorting the  $n$  flows could be completed with time complexity  $O(n \log(n))$ . As proved in [43] (please refer to Theorem 1 in [43]), the time complexity of finding the critical congestion (Algorithm 3) for flow  $f_i$  is  $O(|E_i|)$ . Subtracting the links in  $E_i$  with congestions greater than  $l_i$  takes  $|E_i|$  steps. The time complexity of the Dijkstra's algorithm of finding the path with minimum path congestion takes  $O(|E_i| + |V_i| \log(|V_i|))$  when implemented by

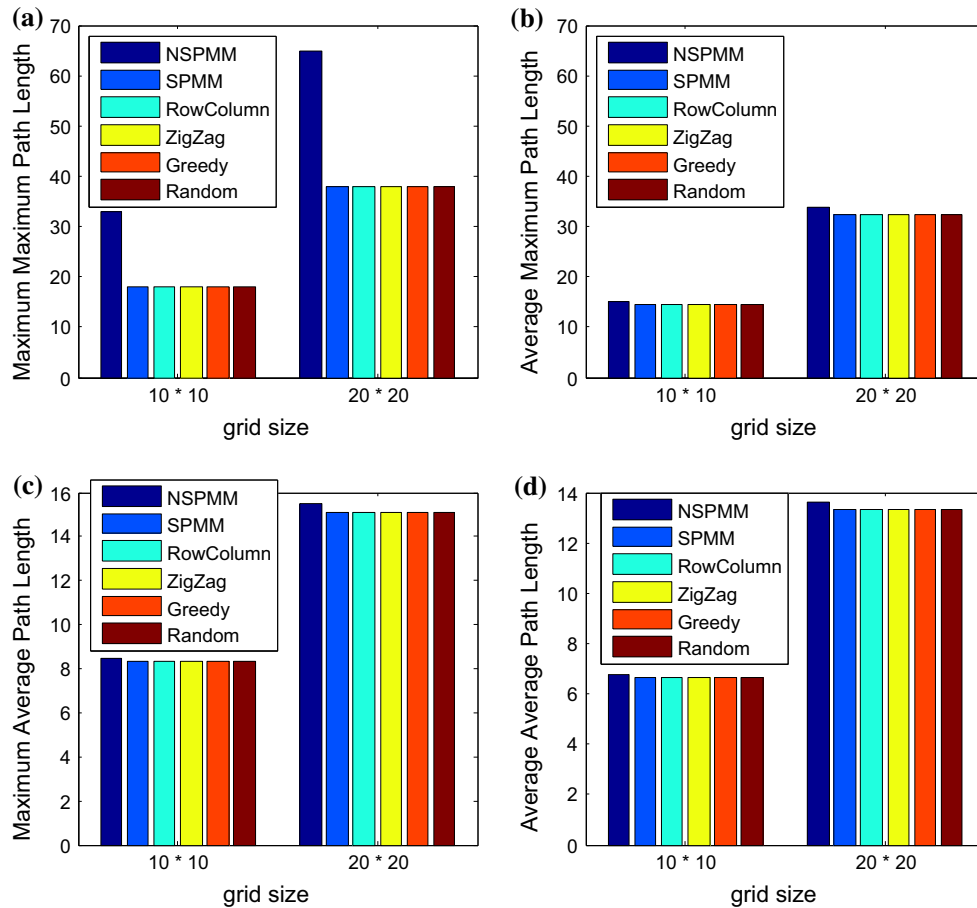


**Fig. 4** Maximum link congestion: partial permutation flows. **a** Maximum maximum link congestion. **b** Average maximum link congestion

**Fig. 5** Distribution of the average link congestion. **a** Average link congestion for the  $10 \times 10$  grid. **b** Average link congestion for the  $20 \times 20$  grid







**Fig. 6** Path length: partial permutation flows. **a** Maximum maximum path length. **b** Average maximum path length. **c** Maximum average path length. **d** Average average path length

priority queue [44]. Thus, for flow  $f_i$ , the time complexity of finding a path is,

$$O(|E_i|) + |E_i| + O(|E_i| + |V_i| \log(|V_i|)),$$

which is also  $O(|E_i| + |V_i| \log(|V_i|))$ . For a  $n_1 \times n_2$  2-dimensional grid, we have that  $|E_i| \leq n_1 \times n_2$  and  $|V_i| \leq n_1 \times n_2$ . There are in total  $n$  flows which implies that the time complexity of the routing algorithm is  $O(n \log(n) + n n_1 n_2 \log(n_1 n_2))$ .  $\square$

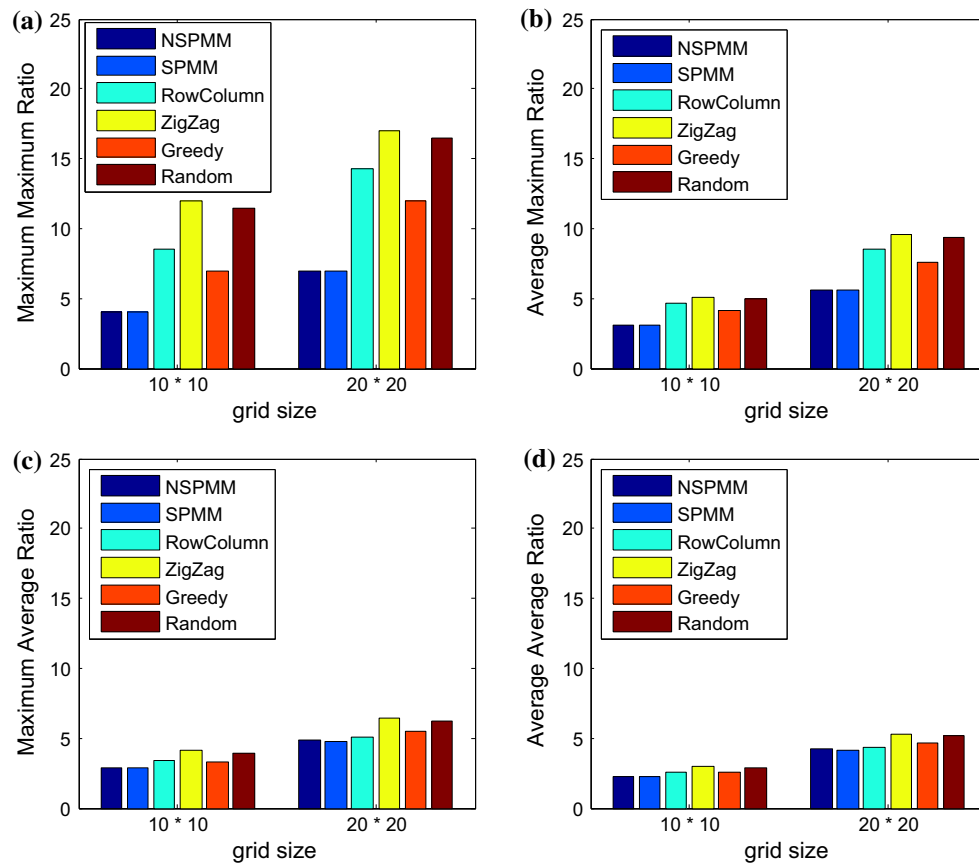
## 5 Performance evaluation

We evaluate the routing algorithms in grid networks in this section. All the algorithms are implemented with JAVA language in Eclipse. The programs are running on a laptop equipped with an Intel(R) Core(TM) i5-2410 M CPU whose frequency is 2.30 GHz. And the size of the RAM is 2.0 GB. The operation system is Windows 7. We test the algorithms in networks with size  $10 \times 10$  and  $20 \times 20$ , the initial configurations of the networks are shown in Table 1.

### 5.1 Scenarios and metrics

We compare the SPMM and NSPMM routing algorithms with other four popular routing algorithms in grid networks, which are Row Column, ZigZag, Random and Greedy routing algorithms. Although we have introduced these four routing algorithms in Sect. 2, we still briefly describe them for a better understanding here.

- In the RowColumn routing algorithm, each packet will be first transmitted along the row to the node in the same column of the destination node, and then along the column to the destination node.
- In the ZigZag routing algorithm, packets will be first transmitted along the row or column to the diagonal node of the destination node, and then transmitted to the destination node using a ZigZag shape path.
- In the Random routing algorithm, each node will select randomly a next-hop node from the legible neighbors which is nearer to the destination nodes.



**Fig. 7** Ratio: partial permutation flows. **a** Maximum maximum ratio. **b** Average maximum ratio. **c** Maximum average ratio. **d** Average average ratio

- In the Greedy routing algorithm, each node will send the packet to the next-hop node which has the smallest congestion.

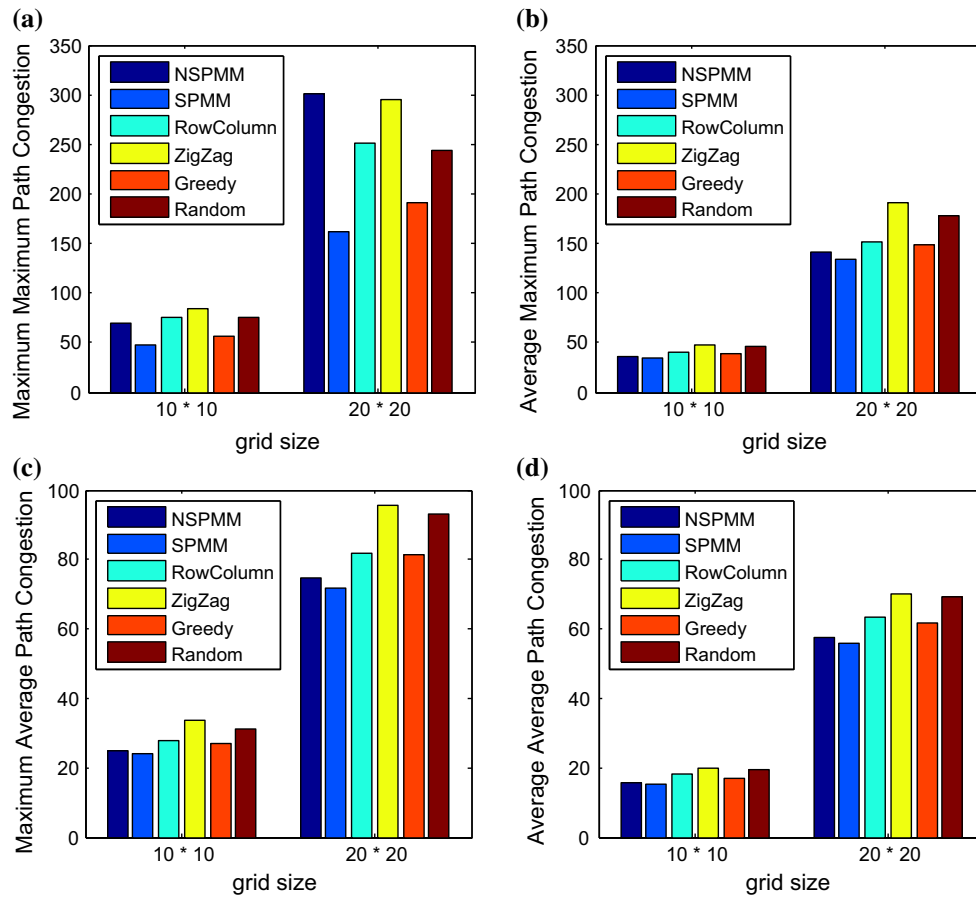
Besides the maximum link congestion and average link congestion, we are also interested in the following metrics.

- *Path Length* For a single path  $p$ , the path length  $h(p) = |\{e|e \in p\}|$  which is the number of links included by the path.
- *Path Congestion* It has been introduced in Sect. 3.
- *Path Congestion to Path Length Ratio* For a single path  $p$ , the path congestion to path length ratio is expressed as  $r(p) = \frac{w(p)}{h(p)}$ .
- *Network delay* The average delay of all the flows is also important.
- *Network throughput* We estimate the network throughput by calculate the number of packets received within a time duration.

For each setting, we run the simulation 20,000 times. Note that, in the expression “Maximum Maximum”, the first “Maximum” means the maximum value among the 20,000 runs, and the second “Maximum” means the maximum value among flows generated in a single instance. Other expressions, such as “Maximum Average”, “Average Maximum” and “Average Maximum” follow the similar explanations.

All the above metrics are evaluated under three traffic patterns. The Partial Permutation Traffic Pattern and General Traffic Pattern have been introduced in the previous section. Here, we introduce the Specific Traffic Pattern ( $W$ ) which we believe will cause very high maximum link congestion of a network. The following construction of the traffic pattern  $W_{n_1, n_2}$  is only suitable for grids with size  $n_1 \times n_2$  where  $n_1 \geq 4$  and  $n_2 \geq 4$ . The concept of constructing the traffic patterns  $W$  can be extended to other situations easily.

The coordinates of the nodes in a grid network have been introduced in Sect. 3. The construction of the specific



**Fig. 8** Path congestion: partial permutation flows. **a** Maximum maximum path congestion. **b** Average maximum path congestion. **c** Maximum average path congestion. **d** Average average path congestion

traffic pattern composes two steps. In step1, we need to construct the four special flows. And in step2, for the remaining nodes, we just construct a symmetric traffic pattern.

1. Step1: construct the special flows.

Now let  $x^* = \lfloor \frac{n_1}{2} \rfloor$  and  $y^* = \lfloor \frac{n_2}{2} \rfloor$ . As shown in Fig. 3, this is the node  $d_5$  with coordinates (3, 3).

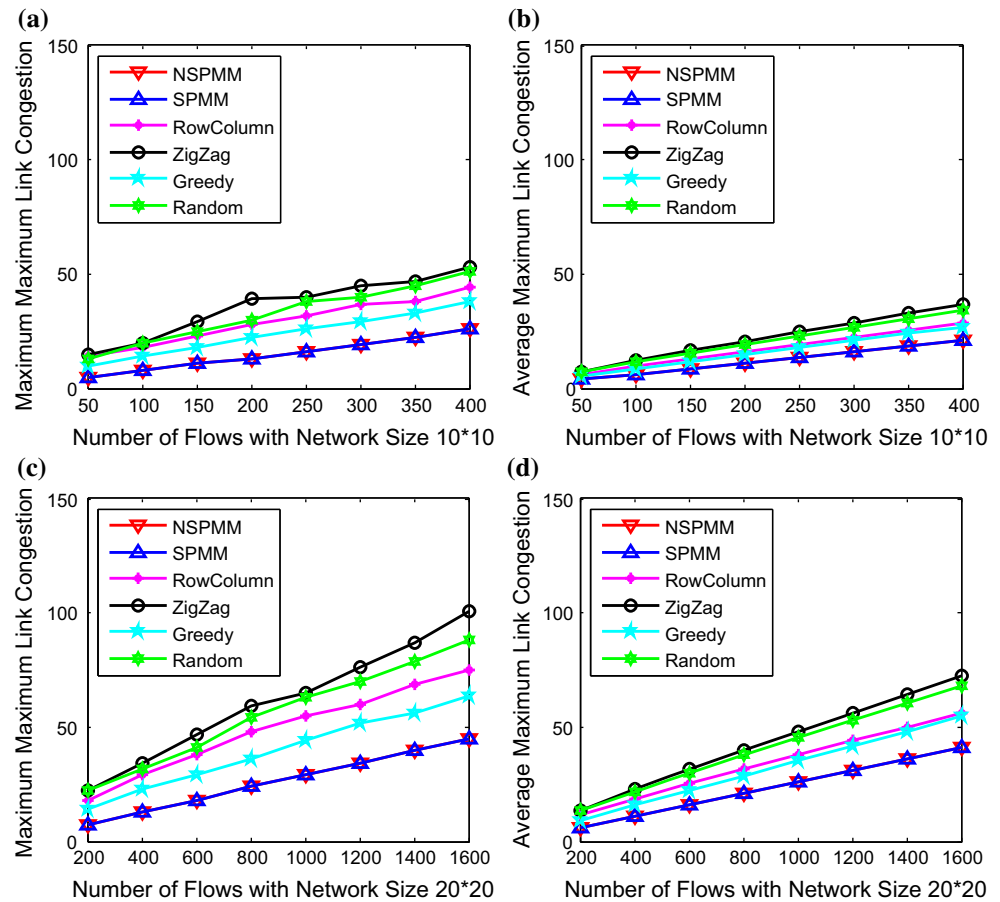
- For node  $(x^*, y^*)$ , the coordinate of the corresponding source node is  $(x^* - 2, y^* - 2)$  (which is node  $s_5$  in Fig. 3);
- For node  $(x^*, y^* - 1)$  (which is node  $s_7$  in Fig. 3), the coordinate of the corresponding destination node is  $(x^* - 2, y^* + 1)$  (which is node  $d_7$  in Fig. 3);
- For node  $(x^* - 1, y^*)$  (which is node  $d_8$  in Fig. 3), the coordinate of the corresponding source node is  $(x^* + 1, y^* - 2)$  (which is node  $s_8$  in Fig. 3);
- For node  $(x^* - 1, y^* - 1)$  (which is node  $s_6$  in Fig. 3), the coordinate of the corresponding source node is  $(x^* + 1, y^* + 1)$  (which is node  $d_6$  in Fig. 3).

2. Step2: construct the symmetric traffic pattern.

For the other nodes, the source and destination nodes are paired according to the following rules.

- When  $n_1$  and  $n_2$  are both even. If a node with coordinate  $(s_i^x, s_i^y) (s_i^y \leq y^*)$  is set to be a source/destination node, then the destination/source node is set to be the one with coordinate  $(n_1 - 1 - s_i^x, n_2 - 1 - s_i^y)$ .
- When one of the values of  $n_1$  and  $n_2$  is odd. Without loss of generality, let  $n_1$  be odd. If a node with coordinate  $(s_i^x, s_i^y) (s_i^x \leq x^*)$  is set to be a source/destination node, then the destination/source node is set to be the one with coordinate  $(n_1 - 2 - s_i^x, n_2 - 1 - s_i^y)$ . After that, the remaining nodes can be arbitrarily paired.
- When both the values of  $n_1$  and  $n_2$  are odd. If a node with coordinate  $(s_i^x, s_i^y) (s_i^x \leq x^* \text{ and } s_i^y \leq n_2 - 1)$  is set to be a source/destination node, then the destination/source node is set to be the one with coordinate  $(n_1 - 2 - s_i^x, n_2 - 2 - s_i^y)$ . After that, the remaining

**Fig. 9** Maximum link congestion: general flows.  
**a** Maximum maximum link congestion:  $10 \times 10$ . **b** Average maximum link congestion:  $10 \times 10$ . **c** Maximum maximum link congestion:  $20 \times 20$ . **d** Average maximum link congestion:  $20 \times 20$



nodes are the ones on the bottom row and right-most column. For a source node among this has the coordinate  $(s_i^x, s_i^y)$ , then the destination node coordinate is  $(s_i^y, s_i^x)$ .

Note that, for a grid  $n_1 \times n_2$  with  $n_1 < 4$  or  $n_2 < 4$ , the construction of a special traffic pattern is a little bit different from the above process. In Fig. 3, the Specific Traffic Pattern of a grid network with size  $6 \times 6$  is given. As shown in Fig. 3, the  $4 \times 4$  center grid is shown in the dashed square. And for all grids, the construction of the most central  $4 \times 4$  grid is the same.

## 5.2 Performance evaluation under the partial permutation traffic pattern

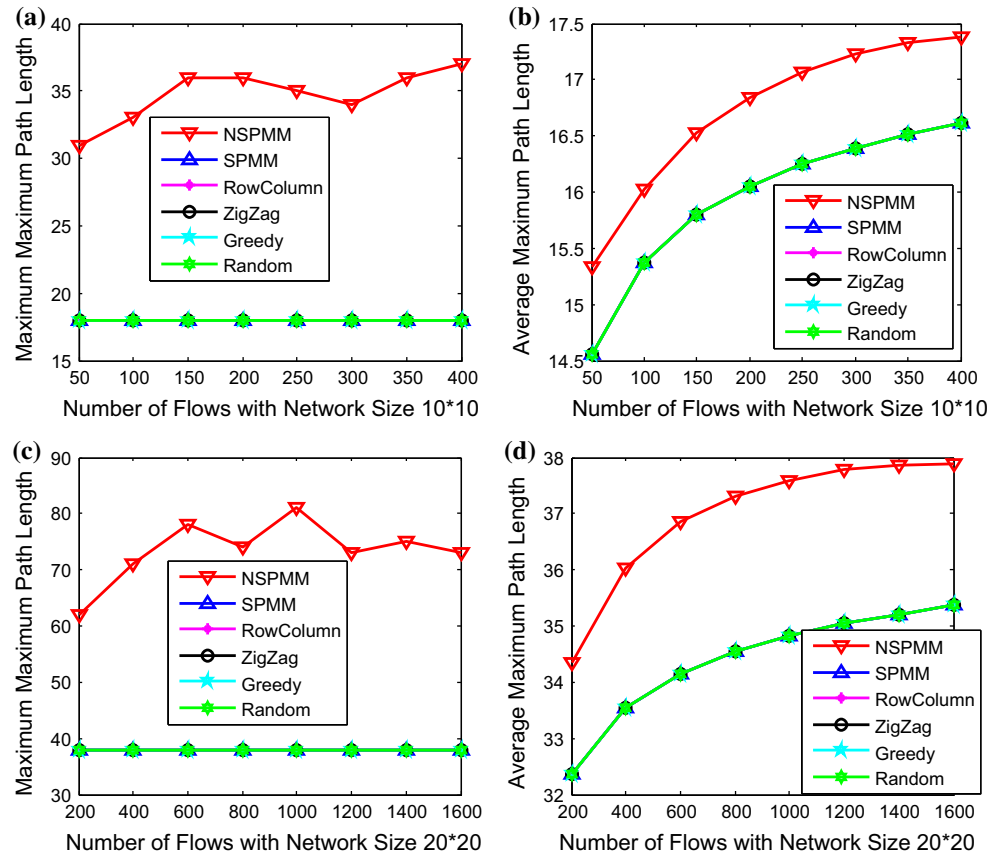
Figure 4 shows the maximum maximum link congestion and average maximum link congestion of six routing algorithms with network sizes  $10 \times 10$  and  $20 \times 20$ . The SPMM and NSPMM routing algorithms show the identical results. The maximum and average maximum link congestions of these two routing algorithms are much smaller than those of the other four routing algorithms. For the

SPMM and NSPMM routing algorithms, the improvement ratios of the maximum maximum link congestions are around 50, 65, 45, and 62 % compared to RowColumn, ZigZag, Greedy, Random routing algorithms, respectively.

The average link congestions of all the shortest path routing algorithms under a certain traffic pattern should be the same. Thus, we only need to compare the average link congestions of the non-shortest path routing algorithm with the shortest path routing algorithms. In Fig. 5, it shows the distribution of the average link congestions in grids with size  $10 \times 10$  and  $20 \times 20$ . The frequencies of SPMM routing algorithm are higher than those of NSPMM routing algorithm for small link congestions. However, for large link congestions, the frequencies of SPMM routing algorithm are smaller than those of NSPMM routing algorithm. It's because of the longer path length generated by NSPMM. The long path lengths may not lead to a larger maximum link congestion (which can be seen from Fig. 4), however they do increase the average link congestion.

Considering that the SPMM, RowColumn, ZigZag, Random and Greedy routing algorithms are all shortest path routing algorithms, the maximum/average path length of each flow under these five routing algorithms are the

**Fig. 10** Maximum path length: general flows. **a** Maximum maximum path length:  $10 \times 10$ . **b** Average maximum path length:  $10 \times 10$ . **c** Maximum maximum path length:  $20 \times 20$ . **d** Average maximum path length:  $20 \times 20$



same, which can be seen in Fig. 6. The differences of path lengths between the non-shortest path routing algorithm and the shortest path routing algorithms can be seen clearly in Fig. 6. Under both the  $10 \times 10$  and  $20 \times 20$  grids, the maximum maximum path lengths, average maximum path lengths, maximum average path lengths and average average path lengths of the NSPMM routing algorithm are larger than those of the five shortest path routing algorithms.

In Fig. 7, we show the path congestion to path length ratios. It's obvious that the maximum maximum, average maximum, maximum average, and average average ratios of the NSPMM and SPMM routing algorithms are smaller than all the other four routing algorithms. The ratio reflects the average link congestion for a path. The smaller the ratio of a path is, the lighter the average link congestion the path has. Still, the differences of the ratios between NSPMM and SPMM routing algorithms are very small. However, they do achieve different performances for some other metrics. This can be seen in Fig. 8(a), the maximum maximum path congestions of the NSPMM routing algorithm under grid networks with sizes  $10 \times 10$  and  $20 \times 20$  are very high. For SPMM routing algorithm, the values in

all the four subfigures are the smallest. Recall the maximum maximum path lengths in Fig. 6(a). The main reason is that the long path lengths lead to the higher path congestions. In a network, let the minimum path congestion be  $l_m$  and the maximum path congestion be  $l_M$ . By increasing a path length by 1, the increasing congestion is at least  $l_m + 1$ , and in some cases, the path congestion may be increased by  $l_M + 1$ . Thus, the path length and network link congestion play important parts in order to achieve smaller path congestion.

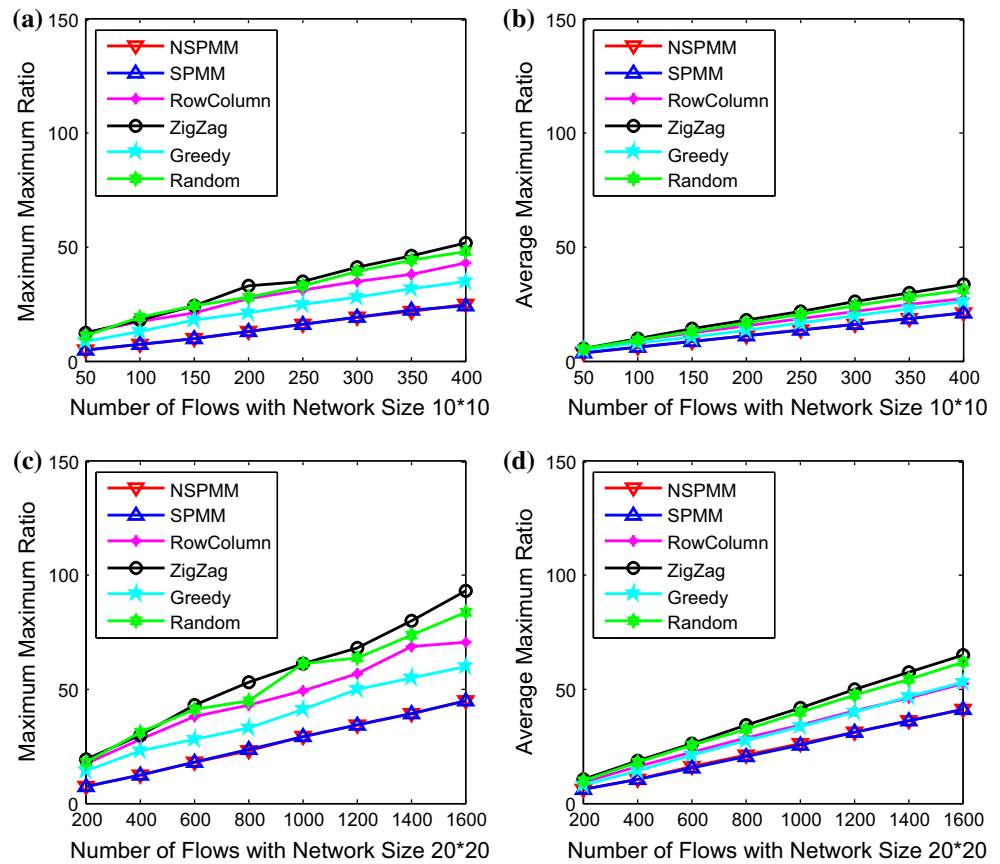
### 5.3 Performance evaluation under the general traffic pattern

For the scenarios with general traffic patterns, we run the simulation for 20,000 times with varying the number of flows. For the  $10 \times 10$  grid, we vary the number of flows from 50 to 400 with an increment of 50. For the  $20 \times 20$  grid, we vary the number of flows from 200 to 1,600 with an increment of 200.

In Fig. 9, we show the maximum maximum link congestions, average maximum link congestions for the grids with sizes  $10 \times 10$  and  $20 \times 20$ . The values increase with



**Fig. 11** Maximum ratio: general flows. **a** Maximum maximum ratio:  $10 \times 10$ . **b** Average maximum ratio:  $10 \times 10$ . **c** Maximum maximum ratio:  $20 \times 20$ . **d** Average maximum ratio:  $20 \times 20$



the increasing of the number of flows. The NSPMM and SPMM routing algorithms achieve the smallest maximum maximum and average maximum link congestions. Like the performances under the partial permutation traffic patterns, the NSPMM and SPMM routing algorithms achieve the identical maximum and average maximum link congestions. This phenomenon motivates us to observe the path lengths under these two routing algorithms. As shown in Fig. 10, in all the four sub-figures, the path lengths of the NSPMM routing algorithm are larger than those of the other five shortest path routing algorithms. And the path lengths of all the shortest path routing algorithms are the same.

For the path congestion to path length ratios shown in Fig. 11, the best performance routing algorithms are the NSPMM and SPMM routing algorithms. Also, differences between the two routing algorithms are not obvious under this metric. However, in Fig. 12, the differences are obvious. The maximum path congestions of the NSPMM routing algorithm become very high. However, the SPMM routing algorithm is still the best under this metric.

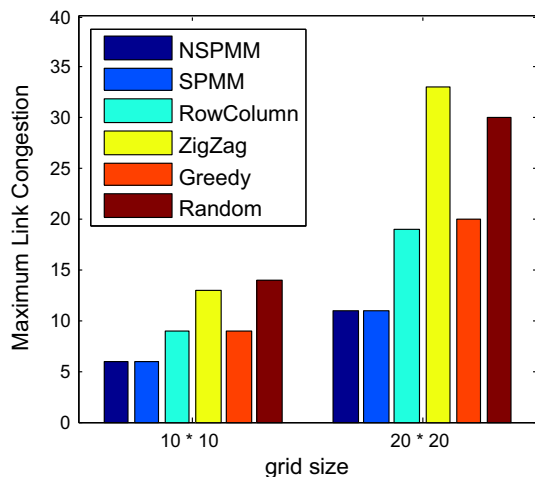
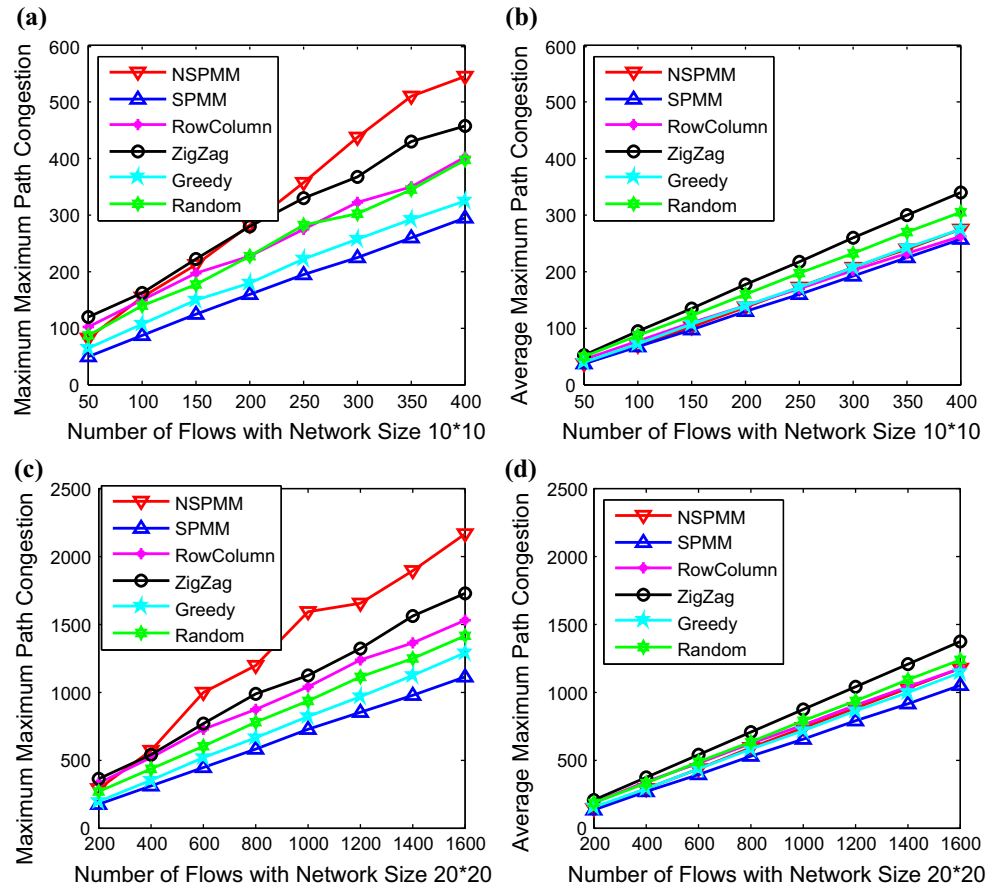
#### 5.4 Performance evaluation under the specific traffic pattern

Now, given the evaluation of the performances of the routing algorithms under the partial permutation and general traffic patterns, we'd like to predict that the SPMM and NSPMM routing algorithms will still achieve the best performances of the maximum link congestions and the path congestion to path length ratios under the Specific Traffic Pattern. The path congestions of the NSPMM routing algorithm should be bad.

The prediction of the maximum link congestions and path congestion to path length ratios are correct, as shown in Figs. 13 and 14. However, for this traffic pattern, the path congestions of the NSPMM are not the worst, as shown in Fig. 15(a) and (b).

Besides the previous observations, there is something interesting about the maximum link congestion which is the main motivation for us to study this traffic pattern. For the  $10 \times 10$  and  $20 \times 20$  grids, the maximum link congestions under this traffic pattern is 6 and 11, respectively. When constructing 20,000 instances of partial permutation

**Fig. 12** Maximum path congestion: general flows.  
**a** Maximum maximum path congestion:  $10 \times 10$ . **b** Average maximum path congestion:  $10 \times 10$ . **c** Maximum maximum path congestion:  $20 \times 20$ . **d** Average maximum path congestion:  $20 \times 20$



**Fig. 13** Link congestion: specific traffic pattern

traffic patterns randomly, the maximum link congestions [refer to Fig. 4(a)] among the 20,000 instances is 5 and 7 for the  $10 \times 10$  and  $20 \times 20$  grids, respectively. Both are

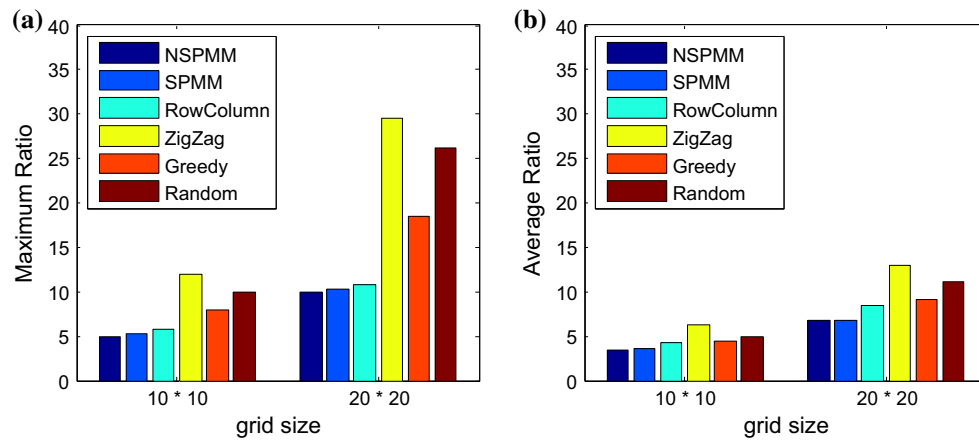
smaller than the corresponding values under the Specific Traffic Pattern.

Although we can't prove it, we strongly believe that the specific traffic pattern is a kind of worst-case traffic pattern for our routing algorithms, which are NSPMM and SPMM. And for a  $n_1 \times n_2$  grid network, the maximum link congestion under this traffic pattern is  $\max\{\lceil \frac{n_1+1}{2} \rceil, \lceil \frac{n_2+1}{2} \rceil\}$ . If the guessing is true,  $\max\{\lceil \frac{n_1+1}{2} \rceil, \lceil \frac{n_2+1}{2} \rceil\}$  will be the upper bound of the maximum link congestion under the NSPMM and SPMM routing algorithms for the Partial Permutation Traffic Patterns.

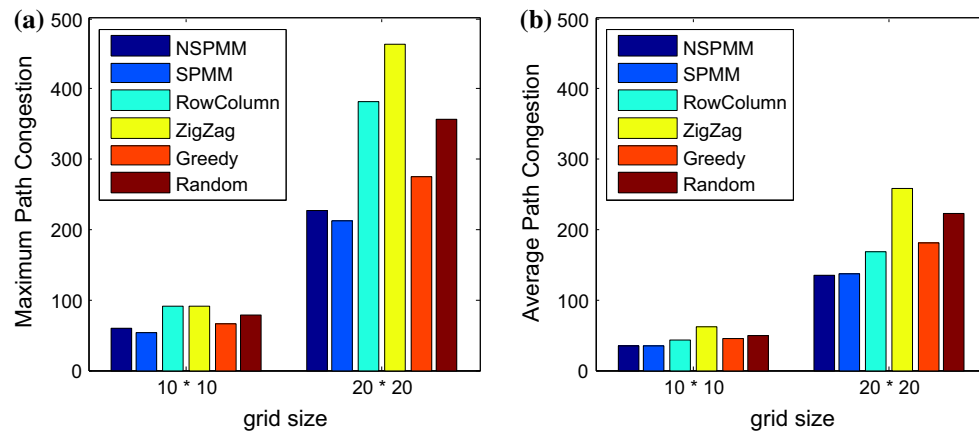
### 5.5 Delay and throughput

Besides the above congestion related performance metrics, delay and throughput are also importance metrics to estimate the network performance. However, the delay and throughput are related to the path length, link congestion of the flow path, and also the scheduling algorithm adopted when transmitting packets.

Here, we choose the simple FCFS (First Come First Serve) scheduling algorithm and run 20,000 instances for



**Fig. 14** Ratio: specific traffic pattern. **a** Maximum ratio. **b** Average ratio



**Fig. 15** Path congestion: specific traffic pattern. **a** Maximum path congestion. **b** Average path congestion

each setting. In this section we only show the results of the general traffic pattern considering that the partial permutation and the specific traffic pattern are special cases of the general traffic pattern.

We show the average delay of the network among all the flows by varying the number of flows in Fig. 16. Also, we choose the network size to be  $10 \times 10$  and  $20 \times 20$ . From the figures, we can see that for the average average delay, the NSPMM and SPMM routing algorithms achieve the shortest delays. And for the average maximum delay, the SPMM routing algorithm achieves the lowest delay, and the NSPMM routing algorithm achieves the second lowest delays.

The throughputs of different routing algorithms are shown in Fig. 17. For the  $10 \times 10$  grid, we choose the delay bound to be 10 slots and 20 slots. For the  $20 \times 20$  grid, the delay bounds are chosen to be 20 slots and 40 slots. When increasing the number of packets, the SPMM

and NSPMM routing algorithms achieve the most number of packets successfully received within the specific delay bound compared to other routing algorithms, except in Fig. 17(c) when the number of flows is 1,600. The reason for this is that the network is in a congested state.

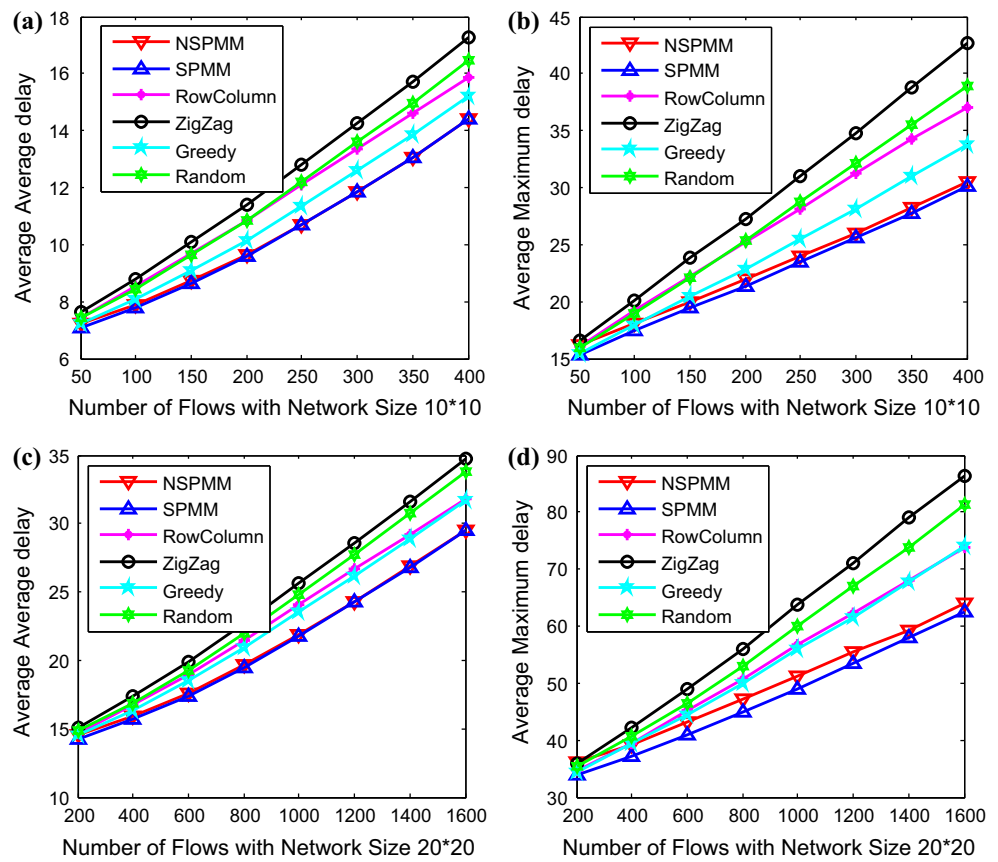
## 5.6 Discussion

### 5.6.1 Maximum link congestion

Under the Partial Permutation Traffic Pattern, a guessing of the upper bound of the maximum link congestion achieved by the NSPMM and SPMM routing algorithms is  $\max\{\lceil \frac{n_1+1}{2} \rceil, \lceil \frac{n_2+1}{2} \rceil\}$  in a  $n_1 \times n_2$  grid network. If the specific traffic pattern is a kind of worst-case traffic pattern, the upper bound will be true.

Through the above numerical results, we know that the NSPMM routing algorithm sacrifices path length to achieve

**Fig. 16** Delay performance.  
**a** Average average delay:  
 $10 \times 10$ . **b** Average maximum  
 delay:  $10 \times 10$ . **c** Average  
 average delay:  $20 \times 20$ .  
**d** Average maximum delay:  
 $20 \times 20$



smaller maximum link congestion. However, this disadvantage can be avoided by the SPMM routing algorithm since it achieves the smallest maximum link congestion and uses only the shortest paths.

### 5.6.2 Algorithm implementation

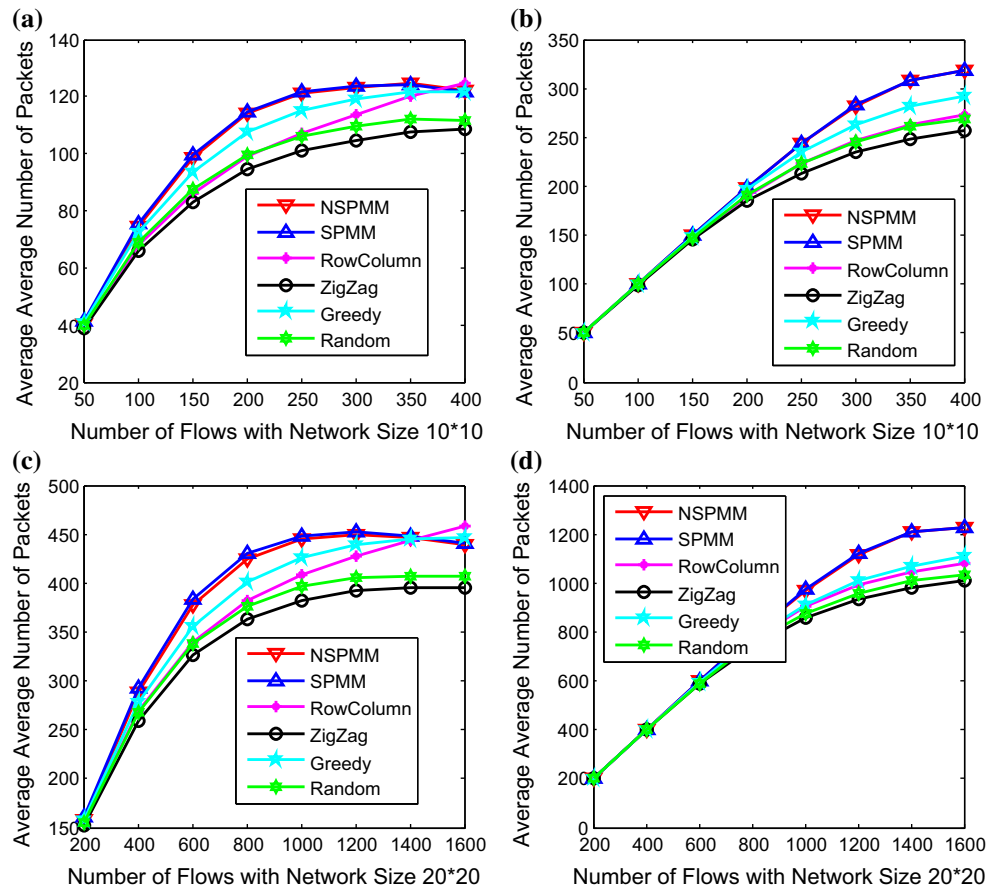
The implementation of the SPMM routing algorithm is much easier than the NSPMM routing algorithm, it can be implemented distributed. The only information that should be informed to the source of each flow is when it should start to seek the routing path. This can be achieved if all the flows first broadcast a packet telling others the number of its shortest paths. According to this information, each flow knows its priority which determines when it can seek the routing path. When a flow finishes the path seeking process, it then broadcasts a packet to inform other flows. If a flow with priority  $i$  receives the routing path completion

notice from the flow with priority  $i - 1$ , it then can start the path seeking process.

## 6 Conclusions

In this paper, we have proposed the NSPMM and SPMM routing algorithms aiming to minimize the maximum link congestion in grid networks. By evaluating these two routing algorithms under the Partial Permutation Traffic Pattern, General Traffic Pattern and the Specific Traffic Pattern, we know that these two routing algorithms can achieve smaller link congestions compared to RowColumn, ZigZag, Random and Greedy routing algorithms. However, the NSPMM routing algorithm has sacrificed the path length and path congestion. And the delay and throughput performance of the SPMM and NSPMM routing algorithms are better than the other four routing algorithms.

**Fig. 17** Throughput performance. **a** Average average number of packets:  $10 \times 10$ , 10 slots. **b** Average average number of packets:  $10 \times 10$ , 20 slots. **c** Average number of packets:  $20 \times 20$ , 20 slots. **d** Average average number of packets:  $20 \times 20$ , 40 slots



Overall, the SPMM routing algorithm is the most efficient routing algorithm.

**Acknowledgments** This work was supported by the National High-tech R&D Program of China (863 Program) (Grant No. 2012AA010904), and the Scientific Research Fund of Sichuan Province, China (Grant Nos. 2013GZ0016, 13ZA0296). And our thanks to the China Scholarship Council (CSC) for the support for the Joint Ph.D. Program.

## References

1. Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., & Xiao, X. (2002). *Overview and principles of internet traffic engineering*. Tech. rep., RFC 3272.
2. Siripongwutikorn, P., Banerjee, S., & Tipper, D. (2002). *Traffic engineering in the internet: A survey of load balanced routing*. White paper.
3. Ravindra Kumar Singh, N. S. C., & Saxena, K. (2012). Load balancing in ip/mps networks: A survey. *Computer Science and Communications*, 4(2), 151–156.
4. Askarian, C., & Beigy, H. (2012). A survey for load balancing in mobile wimax networks. *Advanced Computing: An International Journal*, 3(2), 119–137.
5. Wajgi, D., & Thakur, N. V. (2012). Load balancing algorithms in wireless sensor network: A survey. *International Journal of Computer Networks and Wireless Communications (IJCNCW)*, 2, 456–460.
6. Yao, Y., Cao, Q., & Vasilakos, A. V. (2013). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. In: *MASS* (pp. 182–190).
7. Shen, Zhijie, et al. (2011). Peer-to-peer media streaming: Insights and new developments. *Proceedings of the IEEE*, 99(12), 2089–2109.
8. Suri, P. K., & Kaur, S. (2012). A survey of load balancing algorithms in manet. *Engineering Science and Technology: An International Journal*, 2(3), 495–504.
9. Maheshwari, D., & Nedunchezian, R. (2012). Load balancing in mobile ad hoc networks: A survey. *International Journal of Computer Applications*, 59(16), 44–49.
10. Kelly, F. P., Maulloo, A. K., & Tan, D. K. (1998). Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3), 237–252.
11. Firoiu, V., & Borden, M. (2000). A study of active queue management for congestion control. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1435–1444).
12. Li, Mo, et al. (2013). A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12), 2538–2557.
13. Chen, L., Low, S. H., & Doyle, J. C. (2005). Joint congestion control and media access control design for ad hoc wireless networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (Vol. 3, pp. 2212–2222).
14. Kewei Sha, J. G., & Greve, J. G. (2013). Multipath routing techniques in wireless sensor networks: A survey. *Wireless Personal Communications*, 70(2), 807–829.



15. Radi, M., Dezfouli, B., Bakar, K. A., & Lee, M. (2012). Multipath routing in wireless sensor networks: Survey and research challenges. *Sensors*, 12(1), 650–685.
16. Piratla, N. M. & Jayasumana, A. P. (2006). Reordering of packets due to multipath forwarding—an analysis. In *ICC'06* (Vol. 2, pp. 829–834).
17. Robinson, J. & Knightly, E. W. (2007). A performance study of deployment factors in wireless mesh networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications*. IEEE (pp. 2054–2062).
18. Busch, C., Kannan, R., & Samman, A. (2012). Bottleneck routing games on grids. In *Game Theory for Networks* (Vol. 75 LNCS, pp. 294–307).
19. Leighton, F. T., Maggs, B. M., & Rao, S. B. (1994). Packet routing and job-shop scheduling in  $O$  (congestion + dilation) steps. *Combinatorica*, 14(2), 167–186.
20. Banner, R., & Orda, A. (2007). Bottleneck routing games in communication networks. *IEEE Journal on Selected Areas in Communications*, 25(6), 1173–1179.
21. Busch, C., Kannan, R., & Vasilakos, A. V. (2008) Quality of routing congestion games in wireless sensor networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet*. no. p. 71.
22. Busch, C., & Magdon-Ismail, M. (2009). Atomic routing games on maximum congestion. *Theoretical Computer Science*, 410(36), 3337–3347.
23. Rajgopal Kannan, et al. “Optimal Price of Anarchy of Polynomial and Super-Polynomial Bottleneck Congestion Games.” *GAME-NETS*. pp. 308–320, 2011.
24. Busch, Costas, et al. (2012). Approximating congestion + dilation in networks via ‘quality of routing’ games. *IEEE Transactions on Computers*, 61(9), 1270–1283.
25. Spyropoulos, T., et al. (2010). Routing for disruption tolerant networks: Taxonomy and design. *Wireless Networks*, 16(8), 2349–2370.
26. Zeng, Y., et al. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks*, 19(2), 161–173.
27. Youssef, M., et al. (2014). Routing metrics of cognitive radio networks: A survey. *IEEE Communications Surveys and Tutorials*, 16(1), 92–109.
28. Liu, Y., et al. (2010). Multi-layer clustering routing algorithm for wireless vehicular sensor networks. *IET Communications*, 4(7), 810–816.
29. Chen, Kai, et al. (2011). Survey on routing in data centers: Insights and future directions. *IEEE Network*, 25(4), 6–10.
30. Li, P., Guo, S., Yu, S., & Vasilakos, A. V. (2012). CodePipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding. In *INFOCOM* (pp. 100–108).
31. Demestichas, Panagiotis, et al. (2004). Service configuration and traffic distribution in composite radio environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 34(1), 69–81.
32. Britta, P., Martin, S., & Andreas, W. (2010). Packet routing on the grid. In *Theoretical Informatics—9th Latin American Symposium, Proceedings* (Vol. 6034 LNCS, pp. 120–130).
33. Badr, H. G., & Podar, S. (1989). An optimal shortest-path routing policy for network computers with regular mesh-connected topologies. *IEEE Transactions on Computers*, 38(10), 1362–1371.
34. Weller, T., & Hajek, B. (1994). Comments on “an optimal shortest-path routing policy for network computers with regular mesh-connected topologies. *IEEE Transactions on Computers*, 43(7), 862–863.
35. Wu, J. (1999). Maximum-shortest-path (msp): An optimal routing policy for mesh-connected multicomputers. *IEEE Transactions on Reliability*, 48(3), 247–255.
36. Takatsu, S., Ooshita, F., Kakugawa, H., & Masuzawa, T. (2013). Zigzag: Local-information-based self-optimizing routing in virtual grid networks. In *International Conference on Distributed Computing Systems*, 33rd IEEE (pp. 357–368).
37. Liu, J. W. (2000). *Real-time systems* (pp.115–189). Upper Saddle River, NJ: Prentice Hall PTR.
38. Dhanapala, D. C., Jayasumana, A. P., & Han, Q. (2009). Performance of random routing on grid-based sensor networks. *CCNC*, 2009, 1–5.
39. Rajasekaran, S. (1991). Randomized algorithms for packet routing on the mesh. Technical Reports (CIS). Paper 328. [http://repository.upenn.edu/cis\\_reports/328](http://repository.upenn.edu/cis_reports/328).
40. Busch, C., Magdon-Ismail, M., & Xi, J. (2008). Optimal oblivious path selection on the mesh. *IEEE Transactions on Computers*, 57(5), 660–671.
41. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network flows: Theory, algorithms, and applications* (pp. 649–684). Upper Saddle River, NJ: Prentice Hall PTR.
42. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms* (Vol. 2). Cambridge: MIT press.
43. Kaibel, V., & Peinhardt, M. (2006). On the bottleneck shortest path problem. Technical Reports.
44. Fredman, M. L., & Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3), 596–615.



**Jun Xu** received her Bachelor's degree in the School of Electronic Information, Wuhan University, China, in 2009. She is currently a Ph.D. candidate in the School of Electronic Information, Wuhan University, China. And she is also a visiting scholar in the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, USA, from Sep. 2012 to Mar. 2014. Her research interests are in the area of wireless mesh networks, industry control networks, and real-time and reliability communications.



**Jianfeng Yang** received his Bachelor's degree in Electronic Information in 1998, Master's degree in Signal and Information System in 2003 and Ph.D. degree in Communication and Information System in 2009, all from the school of Electronic Information, Wuhan University, China. He is Currently an Associate Professor of Wuhan University, his research interests are in the areas of Embedded System, Wireless Mesh Networks, Parallel Computing and real-time and Reliability Communications. Also, he served as the committee member of China Computer Federation Technical Committee of Embedded System from 2010.



**Chengcheng Guo** received his Ph.D. degree in the School of Electronic and Information, Wuhan University, China. He received his Bachelor and Master degree in the Computer School of Wuhan University, China. He is currently a professor and a Ph.D. supervisor in the School of Electronic and Information, Wuhan University, China. His research interests are Internet and Communication technology, wireless mesh networks, industry control networks, and real-time and reliability communications.



**Yann-Hang Lee** received his Ph.D. degree in Computer, Information, and Control Engineering from the University of Michigan, Ann Arbor, MI, in 1984. He is currently a professor in the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, USA. Since 2008, he has served as the Program Chair of the Computer Science and Engineering Program in the School. Dr. Lee's research interests are in the areas of real-

time computing, embedded systems, software engineering, distributed systems, and performance evaluation.



**Duo Lu** received his Bachelor's degree in Shanghai Jiao Tong University, China, in 2009. He is currently a Ph.D. student in the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, USA. His research interests are in the area of real-time embedded system and real-time wireless network.