On Rule Placement for Multi-path Routing in Software-Defined Networks

Jie Zhang[1], Deze Zeng[1](✉), Lin Gu[2], Hong Yao[1], and Yuanyuan Fan[1]

[1] School of Computer Science, China University of Geosciences,
Wuhan, Hubei, China
deze@cug.edu.cn

[2] School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, Hubei, China

Abstract. Software Defined Network (SDN) is a newly emerging network architecture with the core concept of separating the control plane and the data plane. A centralized controller is introduced to manage and configure network equipments to realize flexible control of network traffic. SDN technology provides a good platform for application-oriented network innovations to improve network resource utilization, simplify network management, and reduce operating cost. With SDN devices (e.g., OpenFlow switches), routing becomes more flexible by simply changing the contents of flow tables. The flow table is usually implemented in expensive and power-hungry Ternary Content Addressable Memory (TCAM), which is thus capacity-limited. How to optimize the network performance with the consideration of limited TCAM capacity is therefore significant. For example, multi-path routing (MPR) has been widely regarded as a promising method to promote the network performance. However, MPR is at the expense of additional forwarding rule, imposing a heavy burden on the limited flow table. In this paper, we are motivated to investigate an MPR schedule problem with joint consideration of forwarding rule placement. An integer linear programming (ILP) model is formulated to describe this optimization problem. To address the computation complexity, we further design a three-phase heuristic algorithm. Its high efficiency is validated by the fact that it much approaches the optimal solution, according to our extensive simulation studies.

Keywords: Software defined network · Multi-path routing · Rule placement · Optimization

1 Introduction

In traditional network architecture, once the network is deployed, reconfiguration network equipments (e.g., routers, switches, firewall, etc.) to satisfy the changing communication demand is a very complicated task. Now, high stability and high performance of network is not enough to meet the business requirements any more, flexibility and agility are even more crucial. We urgently need a new network architecture innovation to catch up with the increasing demands. Software

S. Guo et al. (Eds.): CollaborateCom 2015, LNICST 163, pp. 59–71, 2016.
DOI: 10.1007/978-3-319-28910-6_6

defined networking (SDN) has emerged as a new paradigm that separates the data plane from the control plane. By such means, the whole network can be administered by centralized controllers using a uniform programming interface as it shields the differences from the underlying network devices. The network thus becomes more flexible and intelligent, making it quite easy to deploy new network protocols [1]. The control plane is completely open. Users can freely customize their network routing strategy according to the application requirements. SDN fills the gap between the application and the network. It is without-doubt the development trend of the future network.

To computer networks, routing is always a major concern and has attracted much attention from scientists and engineers. With the recent prosperousness of computer networks, pioneering researchers found that single-path routing fails to explore the increasing capacity of communication networks. Multi-path routing (MPR), which can find many available paths between communication pairs, was proposed as a promising solution [2]. It has already been proved that MPR is more efficient in improving the utilization of network bandwidth, reducing obstruction, and achieving load balancing. MPR therefore has attracted many interests in the literature [3–5].

It is therefore natural to introduce MPR into SDN for network performance promotion. However, some new challenges are also introduced. In SDN-enabled devices (e.g., OpenFlow switches [6]), the flow tables are made by expensive and power-hungry Ternary Content Aware Memory (TCAM), which is size-limited [3,7]. This limits the number of forwarding rules that can be placed in the flow table. Providing more paths for a network flow usually implies a higher performance, but it also requires more space in the flow table. Therefore, in SDN, MPR shall be carefully scheduled with the consideration of size-limited flow table. Actually, many efforts have been devoted to addressing the limited TCAM size. Most existing studies on forwarding rule optimization, e.g., forwarding rule placement, mainly focus on single-path routing. Taking MPR into consideration, how to jointly schedule the routing and forwarding rule placement is still under-investigated.

In this paper, we are motivated to study the MPR scheduling jointly with the forwarding rule placement in size-limited flow tables. Specially, we are interested in maximize the communication satisfaction with the consideration of communication demands. Our main contributions are as follows.

- To our best knowledge, we are the first to study MPR in SDNs with size-limited flow tables, for communication satisfaction maximization. We build an integer linear programming (ILP) model to describe the optimization problem, with joint consideration of MPR scheduling and forwarding rule placement.
- To address the computation complexity of solving ILP, we further propose a three-phase heuristic algorithm in polynomial time. Through extensive experiment results, the high efficiency of our proposed algorithm is validated.

The rest of this paper is structured as follows. We summarize related work in Sect. 2. We then formulate the problem into ILP in Sect. 3.2. A three-phase heuristic algorithm is presented in Sect. 4. Simulation results are shown in Sect. 5. Finally, we conclude our work in Sect. 6.

2 Related Work

2.1 Multi-path Routing in Traditional Networks

It is widely agreed that MPR has many advantages over single path routing on various aspects such as throughput, load balancing, reliability and so on. In this section, we briefly summarize some representative studies.

Ganjali et al. [8] first evaluate a load balancing problem in MPR. Wei et al. [9] investigate a throughput maximization problem with the goal of reducing bandwidth blocking and improving the average network throughput. They propose a dynamic multi-path provisioning algorithm, which tries to set up dynamic connections with single-path routing in a best-effort manner. Curtis et al. [10] design and evaluate a modification of the OpenFlow model load-balance data center traffic. In [11], it is demonstrated that the proposed flow management reducing overheads, which devolving decisions (as appropriate) to the switches. Al-Fares et al. [12] apply Equal Cost Multipath (ECMP) [2] to efficiently utilize the network resources while achieving network-wide load-balancing. Curtis et al. [13] present Mahout, a low-overhead yet effective traffic management system based on end-host elephant detection. Dasgupta et al. [3] make MPR scheduling based on reliability of the paths. Li et al. [14] study a utility maximization problem with joint consideration of MPR and MAC scheduling. The problem is solved using the primal-dual method. Xu et al. [15] present MPR-based anonymous routing protocol to improve the network effectiveness. Natarajan et al. [16] discuss a location prediction based routing (LPBR) protocol for mobile ad hoc networks (MANETs) and extend it for multicast and MPR.

2.2 Routing Optimization in SDN

Recently, many efforts have been contributed to the field of SDN. Guo et al. [17] consider the effects of large numbers of short flows and massive bursty traffic in data center networks, and design a distributed rate allocation algorithm based on the logistic model under the control-theoretic framework. In their later studies [18–20], they apply game-theoretic framework to the bandwidth allocation problem in data centers. Nemeth et al. [6] demonstrate a large-scale multi-path playground deployed on PlanetLab Europe, which can be used either by experimenters and researchers to test and verify their multipath-related ideas and also by early adopters to enhance their Internet connection [7]. It proves that the MPR is feasible in SDN. Giroire et al. [1] point out that the flow table in OpenFlow switch is implemented using expensive and power hungry TCAM and therefore a limited number of forwarding rules can be stored in SDN switches. Cohen et al. [21] conduct a thorough theoretical study of the bounded path-degree max-flow problem and present approximation algorithms for maximizing network utilization, with the consideration of size-limited flow table.

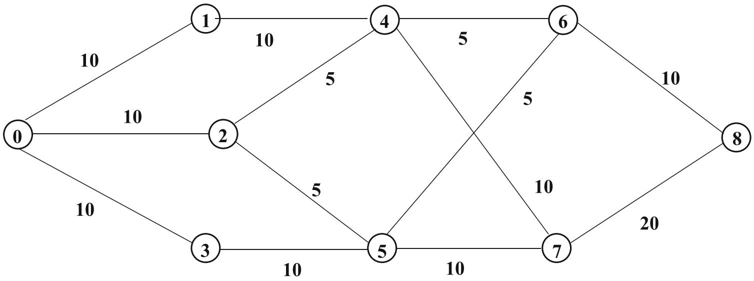


Fig. 1. A network example with 9 nodes and 11 links

3 Problem Formulation

3.1 System Model and Problem Statement

In this work, we consider a network that can be described as a graph $G = (V, E)$, where V is a set of nodes and E denotes the links between these nodes. Figure 1 gives an illustrative network example with 9 nodes and 13 links. Among these nodes, there is a set D of communication pairs. We denote a pair using its source and destination $(s, d) \in D$ where $s, d \in V, s \neq d$. Each pair (s, d) is associated with a traffic demand Π^{sd}. A traffic flow originating from s may go through many intermediate relay nodes (i.e., routers) until reaching destination d. We suppose all the relay nodes are SDN-enabled (e.g., OpenFlow switches). To customize the routing strategy, forwarding rules must be placed at corresponding routers according to the routing requirement. In MPR, the nodes on all forwarding paths for all sub-flows shall be placed with corresponding forwarding rules. Due to the capacity limitation of TCAM, the number of forwarding rules that can be placed on a node is limited. We denote the number of forwarding rules that can be placed on node $v \in V$ as S_v. Besides, the communication links are also capacity-limited. The capacity of link $(u, v) \in E$ is denoted as C_{uv}. For example, we have $C_{01} = 10$ in Fig. 1.

To the requirements of the communication pairs, we are interested in a max-min fairness scheduling which leads to high satisfaction. We define the satisfaction of a pair as the ratio between the achieved throughput and the desired demand. When the provisioned resources are not enough to satisfy the total communication demands, high satisfaction ensures high throughput with max-min fairness. On the other hand, when the achieved throughput is beyond the desired demand, high satisfaction indicates high residual capacity and hence high demand jitter tolerance. Therefore, it is significant to investigate how to maximize the minimum satisfaction of all communication pairs, with joint consideration of MPR and forwarding rule placement in size-limited flow tables.

3.2 Problem Formulation

Based on the system model as described above, we next present an integer linear programming (ILP) model to describe the problem studied in this paper.

Flow Conservation Constraints: As we have known, the traffic flow from node s may go across many different paths consisting of different links to its wanted destination d. We therefore define the flow of communication pair (s, d) on link (u, v) as f_{uv}^{sd}. According to the network flow theory, flow conservation must be reserved at all intermediate relay nodes, except the source and the destination nodes.

As a result, we shall have:

$$\sum_{(u,v)\in E} f_{uv}^{sd} = \sum_{(v,w)\in E} f_{vw}^{sd}, \forall (s,d) \in D, v \in V. \tag{1}$$

Note that the above flow conservation constraints can not be applied to the source and destination nodes, to which we have the following constrains:

$$\sum_{v\in V} f_{sv}^{sd} = F^{sd}, \forall (s,d) \in D, \tag{2}$$

and

$$\sum_{v\in V} f_{vd}^{sd} = F^{sd}, \forall (s,d) \in D, \tag{3}$$

respectively, where float variable F^{sd} indicates the throughput that can be achieved.

Link Capacity Constraints: A traffic flow may go through many different links. While, a link may have sub-flows of many different communication pairs. Nevertheless, the overall flow amount through a link shall not exceed its link capacity. That is,

$$\sum_{(s,d)\in D} f_{uv}^{sd} \le C_{uv}, \forall (u,v) \in E. \tag{4}$$

Flow Table Size Constraints: As we have known, besides link capacity constraints, SDN additionally introduces new constraints on the size of flow tables, indicating the number of forwarding rules that can be stored on the node. Whenever a flow or a sub-flow injects into an SDN node, a corresponding forwarding rule must be placed onto it. Note that no matter how many outgoing sub-flows are scheduled, we always need only one forwarding rule. Let binary variable x_v^{sd} denote whether a forwarding rule of pair $(s, d) \in D$ must be placed on node v. We shall have the following relationship

$$x_v^{sd} = \begin{cases} 1, \text{ if } \exists f_{uv}^{sd} > 0, u \in V, \\ 0, \text{ otherwise.} \end{cases} \forall (s,d) \in D, v \in V. \tag{5}$$

The above relationship can be translated into the following equivalent linear expression as

$$\frac{\sum_{(u,v)\in E} f_{uv}^{sd}}{A} \le x_v^{sd} \le \sum_{(u,v)\in E} f_{uv}^{sd} \cdot A, \forall (s,d) \in D, v \in V, \tag{6}$$

where A is an arbitrary large number.

For each node v, the total number of rules that can be placed onto it shall not exceed its flow table size S_v, i.e.,

$$\sum_{(s,d)\in D} x_v^{sd} \le S_v, \forall v \in V. \tag{7}$$

ILP Formulation: Remember that our objective is to maximize the minimum satisfaction of all communication pairs. It is straightforward to define the satisfaction of a communication pair $(s,d) \in D$ as $\frac{F^{sd}}{\Pi^{sd}}$. Taking all the pairs into consideration, the objective can be described as

$$\max\min : \frac{F^{sd}}{\Pi^{sd}}, \forall (s,d) \in D. \tag{8}$$

In order to simplify the max-min objective into linear form, we define an auxiliary parameter Q with the following constraints

$$Q \le \frac{F^{sd}}{\Pi^{sd}}, \forall (s,d) \in D. \tag{9}$$

Thus, by taking all the constraints discussed above, we obtain an ILP formulation of the max-min fairness problem as

$$\begin{aligned} &\max : Q \\ &\quad \text{s.t.}\ (1) - (4), (6), (7), (9). \end{aligned} \tag{10}$$

4 A Three-Phase Heuristic Algorithm

It is computationally prohibitive to obtain the optimal solution of the ILP problem formulated in last section for large-size SDNs. To tackle this issue, we propose a three-phase polynomial-time heuristic algorithm in this section. In the first phase, by excluding the TCAM size constraints, we first derive the MPR scheduling that can maximize the minimum satisfaction when the flow table is unlimited. We then further take the TCAM size constraints into consideration by checking whether all the nodes satisfy the capacity limitations. Only when no node in a path violates the flow table size constraint, it is feasible; otherwise, it is infeasible and needs to be rescheduled. To adjust these infeasible paths into

Algorithm 1. Find all potential paths

```
Require:
    $G = (V, E)$, $C_{uv}$, $\forall(u, v) \in E$, $D$.
Ensure:
    path set $Allpath$ for every $(s, d)$
Initially, $Allpath = \emptyset$
Exclude constraints (6) and (7) to obtain a LP problem
Solve the LP to obtain $f_{uv}^{sd}, \forall(s, d) \in D, (u, v) \in E$
for all $(s, d) \in D$ do
    for all $(u, v) \in E$ do
        if $f_{uv}^{sd} > 0$ then
            save the $f_{uv}^{sd}$ and link $(u, v)$ to $Allpath$ as a new path ;
            for each path in $Allpath[s, d]$ do
                sort all nodes in $Allpath[s, d]$ ;
            end for
        end if
    end for
end for
```

Algorithm 2. Test the feasibility of all paths

```
Require:
    all paths $Allpath$ for every $(s, d)$ ;
    flow $f_{uv}^{sd}$, $\forall(s, d) \in D, \forall(u, v) \in E$ ;
    flow table size $S_v$, $\forall v \in V$ ;
Ensure:
    dissatisfied paths $dispath$ for every $(s, d)$
    residual flow table capacity $Temp_S$
$Temp_S = S$
for all $(s, d) \in D$ do
    for all path $p^{sd}$ in $Allpath$ do
        if all nodes' rule capacity in this path $> 0$ then
            the path $p^{sd}$ is feasible ;
            for each node $u \in p^{sd}$ do
                $Temp_S[u] = Temp_S[u] - 1$
            end for
        else
            the path $p^{sd}$ is infeasible;
        end if
    end for
end for
```

feasible ones, we discard some sub-flows to make sure that no TCAM size limitation is violated. Further, some sub-flows are merged into the switches with residual flow table capacity to maximize the user satisfaction. The three phases are summarized in Algorithms 1, 2 and 3, respectively.

Phase 1 (Algorithm 1): We first exclude the flow table size constraints by excluding (6) and (7) in ILP. As the integers are only involved in (6) and (7).

Algorithm 3. Reschedule and update the throughput

```
Require:
    all paths $Allpath$ for every $(s, d)$ ;
    infeasible paths $dispath$ for every $(s, d)$ ;
    flow $f_{uv}^{sd} \quad \forall(s, d) \in D, \forall(u, v) \in E$ ;
    Residual flow table size $Temp_S$ ;
Ensure:
    MPR scheduling and rule placement results
1: for all path $p^{sd} \in dispath$ do
2:    if there is a feasible path candidate $p'^{sd}$ in $Allpath$ then
3:       migrate the traffic of path $p^{sd}$ to $p'^{sd}$
4:       update the throughput
5:    else
6:       remove the path $p^{sd}$
7:    end if
8: end for
```

A linear programming (LP) problem is thus obtained. By solving the LP problem, we obtain an MPR scheduling without flow table size constrains. By checking the values of $f_{uv}^{sd}, \forall(s, d) \in D, (u, v) \in E$, we can get whether a link, e.g., $(u, v) \in E$, is involved in the routing of flow, e.g., $(s, d) \in D$, or not. We save the links involved in the routing scheduling (lines 5 to 7). After that, we sort all node in corresponding path to get a full path from line 7 to 8.

Phase 2 (Algorithm 2): In this phase, we try to test whether the nodes in the path involved in MPR, according to the results of Algorithm 1, satisfies the flow table size constraint. As shown in Algorithm 2, we denote the residual flow table size as $Temp_S$, which is initialized with total flow table size S (line 1). By checking whether all the nodes involved in a path are with residual flow table size $Temp_S$ larger than 0, we can judge whether the path is feasible or not. If it is feasible, we subtract the residual flow table size of all the involved nodes by 1 to indicate that one forwarding rule is placed. Otherwise, the path is marked as infeasible and shall be rescheduled.

Phase 3 (Algorithm 3): In this phase, we reschedule all the infeasible paths found out in Phase 2, by migrating their forwarding task to another feasible path in the set found in Phase 1. Similarly, we check both the residual link capacity and flow table size constraints to verify the feasibility of the potential path candidates. If no constraint is violated, we consider the path as feasible (line 3) and migrate the traffic task to this feasible path (line 4). At the same time, the infeasible path is removed from the MPR scheduling (line 7). Otherwise, if no path candidate is found, we have to discard the infeasible path.

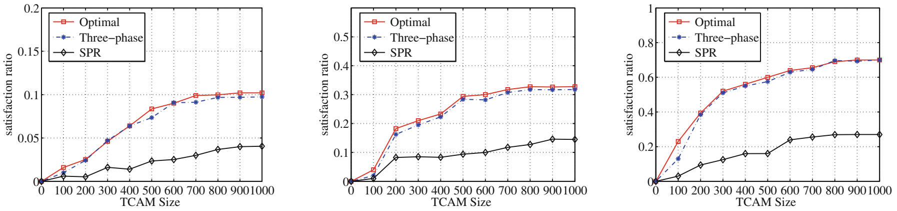
5 Performance Evaluation

To evaluated the performance of the proposed three phase rule placement method ("**Three-phase**"), we present the performance results compared to traditional

SPR algorithm ("**SPR**") and optimal solution ("**Optimal**"). In our experiments, we consider an SDN network consists of 200 nodes with the link-capacity of 1000 and TCAM size of 500 as default. The results proved that our three-phase algorithm significantly out-performances the traditional SPR schema and much approaches the optimal one. To solve our ILP problem and get the optimal results, commercial solver Gurobi[1] is used. We investigate how our algorithm performs and how various parameters affect the communication cost by varying the settings of TCAM Size, link capacity, network size and user demand in each experiment group, receptively.

5.1 On the Effect of TCAM Size

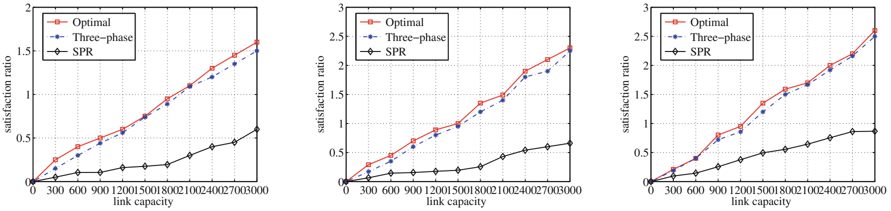
First, we show the max-min satisfaction ratio under different setting of TCAM size from 1 to 1000. Figures 2(a), (b) and (c) show the results with different link capacities of 500, 1000 and 2000, respectively. The experiment results show that the average max-min satisfaction ratio of the network first increases and finally converges with the increasing of TCAM size. This is because for an SDN network with certain user demand, when the TCAM size is small, e.g., from 1 to 100 in Fig. 2(a), the number of forwarding rules stored in flow table is small. Hence the number of flows that can go through a node is limited and the max-min satisfaction ratio is also low. With the increase of TCAM size, e.g., from 100 to 500 in Fig. 2(a), more rules can be stored in the flow table and therefore more flows can be handled at the same time, resulting in an increasing of satisfaction ratio. Finally, when the TCAM size becomes large enough, e.g., from 500 to 1000 in Fig. 2(a), the number of rules is already sufficient for all user demands. While, the link-capacity becomes the bottleneck as it limits the number of flows through a link. In this case, further increasing the TCAM size will not affect the throughput any more and thus all three algorithms converge. Nevertheless, we can always observe from Fig. 2 that our "Three-phase" significantly outperforms "SPR" and performs much close to "Optimal" under any TCAM sizes.



(a) link-capcity=500 (b) link-capcity=1000 (c) link-capcity=2000

Fig. 2. Satisfaction ratio under different TCAM sizes when $|N|$=200

[1] http://www.gurobi.com.



(a) TCAM-size=400 (b) TCAM-size=600 (c) TCAM-size=800

Fig. 3. Satisfaction ratio under different link capacities when $|N|$=500

5.2 On the Effect of Link-Capacity

Then, we investigate the impact of the link-capacity to satisfaction ratio by varying its values from 1 to 3000. Three different TCAM sizes, i.e., 400, 600 and 800, are considered and the results are shown in Fig. 3(a), (b) and (c), respectively. We first observe that the max-min satisfaction ratio of all three algorithms show as an increasing function of link-capacity, under any TCAM sizes. The reason behind such phenomenon is that as the link capacity increases, traffic on each path also increases. Without doubt that this shall lead to a larger max-min satisfaction ratio for a given TCAM size. Besides, the high efficiency of our algorithm is validated once again as we can see that our "Three-phase" algorithm gives a better performance than "SPR" from all three figures.

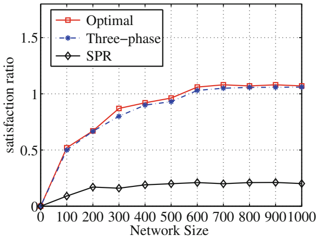


Fig. 4. Satisfaction ratio under different network sizes when TCAM size is 500

5.3 On the Effect of Network Size

Next, we change the network size from 1 to 1000 to investigate how it affects the max-min satisfaction ratio. From Fig. 4, we can see that the satisfaction ratio first increases fast and then converges with network size. The reason for this phenomenon is similar to Fig. 2. Increasing the network size potentially increases the number of TCAMs and the number of available network paths, hence providing more space for rules and more choices of routing paths. Therefore, satisfaction ratio increases with the increase of network size, e.g., from 1 to 750. After the network size is large enough, both the total TCAM size and available network path are already sufficient for current user demand. Under this circumstance, continuously increasing the network size does not take much benefit to the max-min satisfaction ratio. After reaching 750 in Fig. 2, both the link-capacity and TCAM size limit the performance, resulting in a slow increasing of the max-min satisfaction ratio.

5.4 On the Effect of Traffic Demand

Finally, we study the affects of the user traffic demand to the max-min satisfaction ratio. The performance evaluation results are reported in Fig. 5, from which we can see that the max-min satisfaction ratio shows as a decreasing function of the traffic demand. The reason is that, with fixed network topology, TCAM size and link-capacity, the maximum system throughput is also fixed. In this case, increasing the user demand leads to lower satisfaction ratios of all communication pairs, according to (7). Consequently, the max-min satisfaction ratio decreases. Nevertheless, our algorithm is always advantageous over "SPR" algorithm.

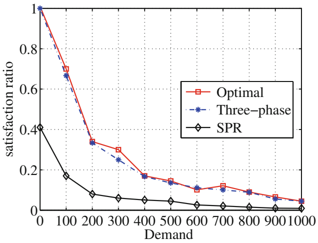


Fig. 5. Satisfaction ratio under different traffic demands

6 Conclusion

In this paper, we study an MPR scheduling problem to maximize the minimum satisfaction ratio of communication pairs in SDNs, with joint consideration of routing scheduling and forwarding rule placement. We first formally formulate the problem into an ILP problem. Then, to tackle the high computational complexity of solving this ILP problem, we propose a three-phase algorithm with polynomial time complexity. Through extensive experiments, we show that our algorithm exhibits substantial advantage over SPR algorithm, and much approaches the performance of the optimal solution.

Acknowledgements. This research was supported by the NSF of China (Grant No. 61402425, 61501412, 61272470, 61502439, 61305087, 61440060), the China Postdoctoral Science Foundation funded project (Grant No. 2014M562086), the Fundamental Research Funds for National University, China University of Geosciences, Wuhan (Grant No. CUG14065, CUGL150829), the Provincial Natural Science Foundation of Hubei (Grant No. 2015CFA065).

References

1. Giroire, F., Moulierac, J., Phan, T.K.: Optimizing rule placement in software-defined networks for energy-aware routing. In: Proceedings of GLOBECOM, pp. 2523–2529. IEEE (2014)
2. Hopps, C.E.: Analysis of an equal-cost multi-path algorithm (2000)
3. Dasgupta, M., Biswas, G.: Design of multi-path data routing algorithm based on network reliability. Comput. Electr. Eng. **38**, 1433–1443 (2012)
4. Cervera, G., Barbeau, M., Garcia-Alfaro, J., Kranakis, E.: A multipath routing strategy to prevent flooding disruption attacks in link state routing protocols for MANETs. J. Netw. Comput. Appl. **36**, 744–755 (2013)
5. Zheng, M., Liang, W., Yu, H., Xiao, Y., Han, J.: Energy-aware utility optimisation for joint multi-path routing and MAC layer retransmission control in TDMA-based wireless sensor networks. Int. J. Sens. Netw. **14**, 120–129 (2013)
6. Németh, F., Sonkoly, B., Csikor, L., Gulyás, A.: A large-scale multipath playground for experimenters and early adopters. In: Proceedings of SIGCOMM, pp. 481–482. ACM (2013)
7. Ford, A., Raiciu, C., Handley, M., Barre, S., Iyengar, J., et al.: Architectural guidelines for multipath TCP development. IETF, Informational RFC **6182**, 1721–2070 (2011)
8. Ganjali, Y., Keshavarzian, A.: Load balancing in ad hoc networks: single-path routing vs. multi-path routing. In: Proceedings of INFOCOM, pp. 1120–1125. IEEE (2004)
9. Lu, W., Zhou, X., Gong, L., Zhang, M., Zhu, Z.: Dynamic multi-path service provisioning under differential delay constraint in elastic optical networks. Commun. Lett. **17**, 158–161 (2013)
10. Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Banerjee, S.: DevoFlow: scaling flow management for high-performance networks. In: Proceedings of SIGCOMM, pp. 254–265. ACM (2011)
11. Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P., Curtis, A.R., Banerjee, S.: Devoflow: cost-effective flow management for high performance enterprise networks. In: Proceedings of SIGCOMM, pp. 158–161. ACM (2010)
12. Al-Fares, M., Radhakrishnan, S., Raghavan, B., Huang, N., Vahdat, A.: Hedera: dynamic flow scheduling for data center networks. In: Proceedings of NSDI, p. 19 (2010)
13. Curtis, A.R., Kim, W., Yalagandula, P.: Mahout: low-overhead datacenter traffic management using end-host-based elephant detection. In: Proceedings of INFOCOM, pp. 1629–1637. IEEE (2011)
14. Li, Y., Zhou, L., Yang, Y., Chao, H.C.: Optimization architecture for joint multipath routing and scheduling in wireless mesh networks. Math. Comput. Model. **53**, 458–470 (2011)
15. Xu, L., Zhao, W., Jiang, L., Jin, J., Gui, N.: Multi-path anonymous on demand routing protocol. In: Proceedings of IMCCC, pp. 858–863. IEEE (2013)
16. Meghanathan, N.: A location prediction based routing protocol and its extensions for multicast and multi-path routing in mobile ad hoc networks. Ad Hoc Netw. **9**, 1104–1126 (2011)
17. Guo, J., Liu, F., Huang, X., Lui, J., Hu, M., Gao, Q., Jin, H.: On efficient bandwidth allocation for traffic variability in datacenters. In: Proceedings of INFOCOM, pp. 1572–1580. IEEE (2014)

18. Guo, J., Liu, F., Tang, H., Lian, Y., Jin, H., Lui, J.: Falloc: fair network bandwidth allocation in iaas datacenters via a bargaining game approach. In: Proceedings of ICNP, pp. 1–10. IEEE (2013)
19. Guo, J., Liu, F., Zeng, D., Lui, J., Jin, H.: A cooperative game based allocation for sharing data center networks. In: Proceedings of INFOCOM, pp. 2139–2147. IEEE (2013)
20. Guo, J., Liu, F., Lui, J., Jin, H.J.: Fair network bandwidth allocation in iaas datacenters via a cooperative game approach. IEEE/ACM Trans. Netw. (2015)
21. Cohen, R., Lewin-Eytan, L., Naor, J.S., Raz, D.: On the effect of forwarding table size on SDN network utilization. In: Proceedings of INFOCOM, pp. 1734–1742. IEEE (2014)