Achieving Efficient Routing in Reconfigurable DCNs

ZHENJIE YANG, Tsinghua University, China
YONG CUI*, Tsinghua University, China
SHIHAN XIAO, Huawei Technologies, China
XIN WANG, Stony Brook University, USA
MINMING LI, City University of Hong Kong, China
CHUMING LI, Tsinghua University, China
YADONG LIU, Tsinghua University, China

Heavy and highly dynamic traffic demands in today's data center networks (DCNs) pose great challenges to efficient traffic engineering. With gigabit bandwidth, wireless communication technologies, such as free space optics and 60GHz wireless, are promising to augment DCNs and enable efficient traffic engineering. Complementary to the emerging reconfigurable architectures, we aim to achieve efficient routing and effectively balance the load with the performance guarantee. We derive a general interference model and propose a decomposition technique with proven performance guarantee and solve the load balancing problem in reconfigurable DCNs. In addition, we propose two solutions, WiRo and OFS, to flexibly reconfigure network topology and enable hybrid-routing with paths consisting of both stable wired links and flexible wireless links with different methods. Our measurement-facilitated and trace-driven simulations demonstrate that our solutions outperform existing flow scheduling algorithms with the average throughput of large flows increased by up to 190% and the average completion time reduced by up to 72.6%. Meanwhile, the average completion time of small flows is reduced by up to 64.5%.

CCS Concepts: • **Networks** → **Data center networks**; *Topology analysis and generation*; • **Theory of computation** → **Routing and network design problems**.

Additional Key Words and Phrases: wireless communication, load balancing

ACM Reference Format:
Zhenjie Yang, Yong Cui, Shihan Xiao, Xin Wang, Minming Li, Chuming Li, and Yadong Liu. 2019. Achieving Efficient Routing in Reconfigurable DCNs. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 3, Article 47 (December 2019), 30 pages. https://doi.org/10.1145/XXXXXXX

*Yong Cui is the corresponding author.

Authors' addresses: Zhenjie Yang, yangzj15@mails.tsinghua.edu.cn, Tsinghua University, China; Yong Cui, cuiyong@tsinghua.edu.cn, Tsinghua University, China; Shihan Xiao, xiaoshihan@huawei.com, Huawei Technologies, China; Xin Wang, x.wang@stonybrook.edu, Stony Brook University, USA; Minming Li, minming.li@cityu.edu.hk, City University of Hong Kong, China; Chuming Li, lichuming.lcm@gmail.com, Tsinghua University, China; Yadong Liu, liuyd17@mails.tsinghua.edu.cn, Tsinghua University, China.

2476-1249/2019/12-ART47 $15.00
https://doi.org/10.1145/XXXXXXX

1 INTRODUCTION

With the fast growth of cloud services and network scales, the heavy and highly dynamic traffic demands pose great challenges to the efficient traffic engineering in today's data center networks (DCNs) [43]. The DCN flows can be broadly classified into two main categories: delay-sensitive *small flows* (e.g., queries or real-time small messages) and throughput-sensitive *large flows* (e.g., the backup traffic). In general, more than 80% flows in data centers are small flows, while the majority of the traffic volume is contributed by the top 10% large flows [4, 9]. To handle the mixed traffic, today's data centers [2, 24] generally follow the tree-based topologies (e.g., fat-tree) and take the load-agnostic routing strategies based on random path selection (e.g., ECMP[1]) [24, 41]. Although it is applicable for routing small flows which are highly random, these strategies are likely to route several large flows through the same output link and lead to long-lived congestions [3, 11]. With the limited switch buffer occupied by large flows for a long time, small flows are reported to experience one order of magnitude larger delay, which compromises the performance of DCNs and makes the users suffer [4].

To address the above problem, many research efforts turn to the rapidly-developing wireless communication technologies such as 60GHz wireless, free-space optics (FSO) and optical switches, which show great potential in augmenting DCNs with full flexibility and Gigabit per-link capacity at low cost [19, 25, 26, 46]. Taking advantage of reconfigurable data center architectures, demand-aware networking techniques have been studied to achieve high-performance data transmission in data centers [5–7, 20, 21, 40].

The performance of existing routing solutions in reconfigurable DCNs is severely limited by the new challenges resulted from the wireless communications, especially at the high frequency band. Despite its low installation cost and high flexibility in configuration, compared to FSO and optical switch, 60GHz wireless technology involves higher complexity in design with its need of considering interference [16] and short-distance transmission [52]. Thus we illustrate our scheme with 60GHz wireless technology as an example, while our design is general and can be adapted to other techniques that can reconfigure the network topology. 60GHz wireless links have some special features. First, they are highly directional but reconfigurable for more flexible communications [16, 52]. Second, due to the fast attenuation of high-frequency wireless signals, the transmission distance of a 60GHz wireless link is just several meters [25, 42]. Third, as racks are small in size and densely located in DCNs, the interference among wireless links limits their availability and bandwidth stability [15, 53]. This further reduces the chance of concurrent transmissions.

The initial efforts that apply wireless techniques to data centers generally assume one of the two strategies: testbed experiments to understand the wireless features in DCNs [25, 42], or theoretical analysis without considering the practical wireless factors such as the interference impacts [27]. Instead, we build our model based on our measurement studies and take into account the practical constraints in our problem to provide better performance guarantee. As non-segregated routing has many advantages over the segregated one [2] [20, 21], we take the non-segregated routing that large flows can be transmitted over both wired and wireless links.

[1]The ECMP (i.e., Equal-Cost Multi-Path) strategy routes a flow by randomly assigning one path from all the equal-length shortest-paths.

[2]In segregated routing, large flows are routed on direct reconfigurable links while the remaining traffic is left to the static wired links.

Minimizing the maximum congestion level of all links is a desirable feature of DCNs [25, 27]. Our goal in this paper is to balance the load in reconfigurable DCNs with algorithms that have proven performance bound. In response to practical traffic loads where a small number of large flows account for the majority of total traffic volume, our solutions flexibly reconfigure the wireless links and efficiently route large flows to achieve network-wide load balancing, taking into account the wireless interference. The main contributions of our work are summarized as follows.

First, we analyze the traffic traces of real data centers and compare different routing strategies to reveal the feasibility of achieving efficient routing in reconfigurable DCNs by scheduling large flows. We formulate the joint optimization problem of hybrid routing with practical interference constraints in reconfigurable DCNs and prove its NP-hardness. Then we develop a decomposition technique with proven performance guarantee to divide the problem into two easier subproblems.

Second, we design two flow scheduling solutions, WiRo and OFS, to achieve efficient routing in reconfigurable DCNs with different workloads. The two solutions perform network-wide load balancing with two major techniques, reconfiguring network topology with the flexible setup of wireless links in response to traffic demand, and transmitting large flows with hybrid routing over both flexible wireless links and stable wired links. Compared with existing methods for approximating independent-set polytope, our solutions reduce the approximation ratio by up to 87%, thanks to our better modeling of interference of 60GHz antenna thus the topology of data centers. We show that our performance-guaranteed hybrid routing solutions can be extended to apply in other important interference models such as that of the recent 3D beamforming technology [16, 52, 53].

Third, we measure the wireless parameters of interest in the realistic 60GHz platform, based on which we conduct trace-based evaluations to verify the efficiency of our solutions under various network settings. Extensive simulation results show that, compared to existing flow scheduling methods in DCNs, our solutions can increase the average throughput of large flows by up to 190% and reduce their average completion time by up to 72.6%. The average completion time of small flows is reduced by up to 64.5% in the meanwhile. The code is open source and is available at [].

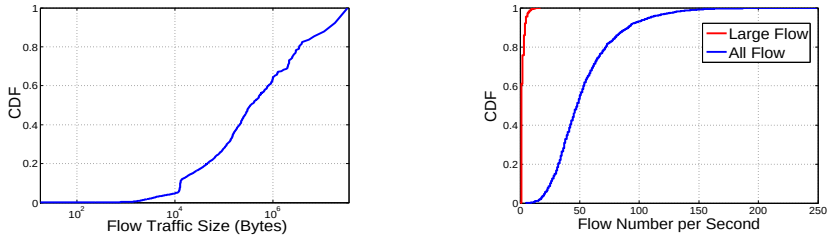
The remainder of this paper is organized as follows: we introduce the problem formulation and decomposition in Section 2. We present WiRo in Section 3 and OFS in Section 4. We evaluate our solutions in Section 5. Then we discuss the deployment issues in Section 6 and introduce the related work in Section 7. Finally, we conclude our work in Section 8.

2 PROBLEM FORMULATION AND DECOMPOSITION

In this section, we first analyze the traffic traces of data centers to provide the motivation of our work, we then give the definitions, notations and our model for the wireless interference. Finally, we formulate our problem and present a decomposition solution to address it.

2.1 Trace analysis and routing strategies

We first analyze the traffic from the one-hour data traces (containing about one million flow entries) of the university data centers provided by [10]. Fig. 1 shows the distribution of traffic sizes and flow arrival rates. In Fig. 1a, we can see that there is a long tail for the distribution of flow traffic size up to 100MBytes. The top 10% of large flows account for 61% of the total traffic volume, which confirms the previous finding that the top 10% of large flows contribute to the majority of the traffic volume in DCNs [9, 34]. In Fig. 1b, we see that the distribution of flow number per second has a long tail with about 8% flows arriving



(a) CDF of flow traffic size (b) CDF of flow arrival rate

Fig. 1. Analysis of flow traces in data centers

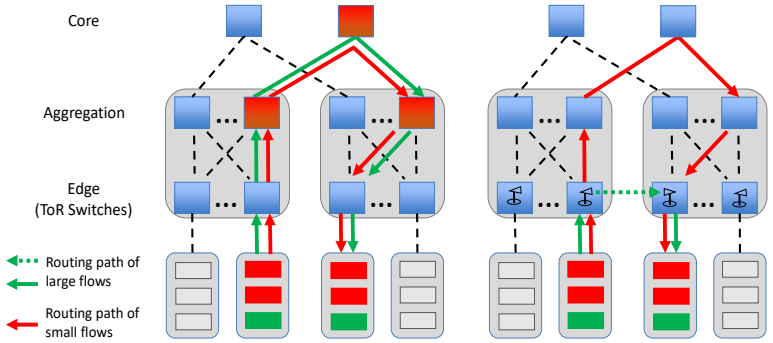


Fig. 2. Different routing strategies in data centers

within an interval less than 0.01s. Another interesting finding is that, by considering the flows that have the traffic sizes larger than 100KBytes as large flows, there are at most 8 large flows per second, while under 90% cases, there are at most 4 large flows per second. This indicates that a periodic scheduling of large flows in DCNs is feasible, as the arrival rate of large flows is very low compared to that of all the flows.

We compare different routing strategies, *wired-only routing* and *hybrid routing*, in a simple tree-based DCN. With wired-only routing in Fig. 2a, all small flows and large flows are forwarded by switches located at aggregation layer and core layer. Due to the long occupancies of switch buffers with large flows, small flows suffer from long delays, which compromises the online service experiences of users. This problem can be alleviated with the use of hybrid routing. As shown in Fig. 2b, large flows can be forwarded through both wired and wireless links between racks, and small flows are forwarded through wired links as done with the above wired-only routing strategy. The paths of large flows are cut short with the use of wireless links, reducing the load of higher layer switches. Without the need of competing with large flows for routing, the delays of small flows are reduced naturally.

2.2 Definitions and notations

Benefited from the emerging software defined network techniques, it is feasible for data center operators to adopt a central controller to determine the setup of wireless links and the routing paths for large flows, while the small flows are served by the default ECMP strategy. We denote the topology of a wireless data center network as a graph $G(V, E)$, where V denotes the set of devices, such as switch and server. The set of links E connect the devices, and is divided into the wired set E_w and the wireless set E_s. Wireless radios are

deployed on the top of racks and wireless links are only built between edge switches (i.e, the ToR switches) [25]. We have $E_w \cap E_s = \emptyset$ and $E = E_w \cup E_s$. Each link $e_{ij} \in E$ connecting v_i and v_j has a link capacity of $\mathcal{C}_{ij}$. We denote the number of devices and the number of links as $|V|$ and $|E|$ for simplicity.

For a given graph G, the wired link set E_w is fixed, while the wireless link set E_s can be changed on demand. We denote the link set $\overline{E}_s$ as all the wireless links *available* for routing if not considering the wireless interference, and $E_s \subset \overline{E}_s$. Specifically, the wireless links that are longer than the maximum transmission distance (e.g., 10m [25]) or physically blocked by other obstacles [52] are always considered as *unavailable*, i.e., they are excluded from $\overline{E}_s$. To guarantee the transmission performance of wireless links, only the links whose bandwidth is larger than a given threshold (e.g., 1Gbps) are considered as available in this paper.

2.3 Wireless interference

Conventional models for the wireless interference are generally classified into two types: the protocol interference model (PIM) and the physical interference model (PHY) [32]. In the PIM model, a transmission on link e_{ij} is considered to be successful if and only if node v_j is not within the interference range of other transmissions. Different from the PIM model, the PHY model allows the transmission over a wireless link as long as the aggregate interference from other nodes is low enough. It requires the controller to deal with varying link capacities very carefully based on the real-time transmissions of neighboring links. The dynamic arrivals of large flows would make the interference relationship change and complicated, which would introduce large overhead if not completely infeasible to re-examine the scheduling plan to maximize the throughput. On the other hand, the PIM model conservatively considers the interference condition, and a central controller only needs to maintain the non-interfered links with a low overhead. Since it is important for us to consider a more stable wireless condition for throughput-sensitive large flows in DCNs, we apply the PIM model in our formulation. We will show that our solution can be easily extended to deal with the PHY model while still keeping a proven performance guarantee.

Generally, antennas are deployed on the top of racks and they can communicate with each other if there is no interference link. For conventional omnidirectional antennas, the interference range has a circular shape. To embrace the gigabit-level bandwidth requirement in DCNs, the 60GHz wireless technology is proposed, which uses the highly directional antenna to generate a fan-shape interference range [52]. To make the model general without relying on specific antenna types, we denote an *interference link set* S_e of a wireless link e as the links that will interfere with the transmission on e, and construct a *conflict graph* $G_c(V_c, E_c)$ to describe the interference relations among all wireless links in $\overline{E}_s$. Each wireless link $e \in \overline{E}_s$ is a vertex in G_c, and an edge (e_1, e_2) is added between any two vertices e_1 and e_2 if and only if they conflict with each other, i.e., either $e_1 \in S_{e_2}$ or $e_2 \in S_{e_1}$. The conflict relationship between wireless links can be obtained with offline measurement based on the types and locations of antennas in DCN, and would not change when the rack deployment is fixed [25]. We denote an *independent set* (IS) in a conflict graph $G_c(V_c, E_c)$ as a vertex subset of V_c where there is no edge connecting any two vertices. Hence a feasible solution for setting up wireless links in G is to select an IS in the conflict graph G_c which satisfies:

$$\gamma_{ij} = 0 \text{ or } 1, \ \forall e_{ij} \in \overline{E}_s \tag{1}$$

$$\gamma_{ij} + \gamma_{uv} \leq 1, \ \forall (e_{ij}, e_{uv}) \in E_c \tag{2}$$

where γ_{ij} denotes whether the wireless link e_{ij} is selected to be built. Constraints (2) will be violated if any two wireless links selected to be built interfere with each other.

2.4 Problem formulation

As minimizing the maximum congestion level in all links is a desirable feature of DCNs [25, 27], the objective of our work is to minimize the maximum link utilization of the entire network during each scheduling period, so we can ensure load-balanced transmission and reduce the transmission delay of flows. The controller input is $|F|$ large flows and each flow $f_k \in F$ is attached with a flow demand d^k. We denote the source and destination node of f_k as s_k and t_k. Let a binary variable x_{ij}^k indicate whether a flow f_k routes through the link e_{ij}. Splitting a flow into multiple sub-flows and transferring them in different routing paths require the data center operators to modify the transportation protocol, which will inevitably result in reordering problem. In this work, we solve the problem with *unsplittable* flows, i.e., a flow can only be routed on a single path. We use $\mathcal{C}_{ij}$ to denote the link capacity of link e_{ij}. Hence the link utilization of link $e_{ij} \in E$ is $\sum_{f_k \in F} x_{ij}^k d^k / \mathcal{C}_{ij}$. For a feasible routing solution, a wireless link must be built if there is at least one flow routed through it, hence we have:

$$\gamma_{ij} \geq x_{ij}^k, \ \forall e_{ij} \in \overline{E}_s, \ f_k \in F \tag{3}$$

Let λ denote the *maximum link utilization* in the network. Our optimization problem $\mathcal{P}_0$ is formulated as follows:

$$\min \ \lambda$$

s.t.

$$\sum\nolimits_{f_k \in F} d^k x_{ij}^k \leq \lambda \mathcal{C}_{ij}, \ \forall e_{ij} \in E \tag{4}$$

$$\sum\nolimits_{e_{ij} \in E} x_{ij}^k = \sum\nolimits_{e_{ji} \in E} x_{ji}^k, \ \text{if } v_i \notin \{s_k, t_k\} \tag{5}$$

$$\sum\nolimits_{e_{ij} \in E} x_{ij}^k = 1, \ \text{if } v_i = s_k \tag{6}$$

$$\sum\nolimits_{e_{ji} \in E} x_{ji}^k = 1, \ \text{if } v_i = t_k \tag{7}$$

$$x_{ij}^k = 0 \text{ or } 1, \ \forall e_{ij} \in E, \ f_k \in F \tag{8}$$

Constraints (1)(2)(3)

Constraints (4) are the link capacity constraints and (5)(6)(7) are for flow conservation. Constraints (8) ensure that the flow is unsplittable. Constraints (1)(2)(3) are appended to take into account the wireless interference. By solving this optimization problem, the controller outputs the routing paths selected for each flow $f_k \in F$ and an IS for the setup of wireless links.

There are two types of NP-hardness embedded in $\mathcal{P}_0$. The first hardness is derived from the unsplittable flow property, which is required by the high performance of flows in DCNs [41]. We can reduce the integer partition problem to $\mathcal{P}_0$ in polynomial time, and have the following theorem:

Theorem 1. *$\mathcal{P}_0$ is NP-hard.*

The detailed proof can be found in Appendix A.1. The second hardness comes from the complex wireless interference caused by the high flexibility of forming 60GHz wireless links [52] in DCNs. We need to select the optimal IS to set up the non-interfered wireless links in the wireless conflict graph. However, finding all the ISs is NP-complete in general [22]. Moreover, the small angle of interference for 60GHz links would lead to exponentially many ISs in DCNs.

Therefore, we develop a decomposition technique to split the mixed hardness. The original problem $\mathcal{P}_0$ is decomposed into two sub-problems $\mathcal{P}_1$ and $\mathcal{P}_2$ as shown in Table 1. First, by

relaxing the unsplittable flow constraint (8) in $\mathcal{P}_0$ to $0 \leq x_{ij}^k \leq 1$, we have the subproblem $\mathcal{P}_1$: a splittable flow problem which gives the optimal solution if a flow is allowed to be split over multiple routing paths to provide a lower bound for $\mathcal{P}_0$. Second, by removing the wireless interference constraints (1)(2)(3) in $\mathcal{P}_0$, we have the subproblem $\mathcal{P}_2$.

The motivation of decomposing the original problem $\mathcal{P}_0$ is to apply a relaxed $\mathcal{P}_1$ to find a *nice* IS among the flexible wireless links without considering the *unsplittable flow* property. We can then fix this IS thus the wireless links and solve $\mathcal{P}_2$ to find the exact routing paths without considering the wireless flexibility and interference.

Since the flow variables x_{ij}^k and wireless variables γ_{ij} are closely coupled with each other in constraints (3), it is non-trivial to decompose the original problem with a guaranteed approximation ratio. In the following theorem, we illustrate the merit of our decomposition technique. For a minimization problem, we denote a *ρ-approximation* algorithm as the one that can achieve a solution within ρ times the optimal solution ($\rho \geq 1$). Specially, we call an approximation algorithm for an integer programming (IP) minimization problem *ρ-relaxed* if it can achieve a solution within ρ times the optimal solution of its LP relaxation. Based on the combination guarantee in [17] and [49], we have the following theorem:

THEOREM 2. *Suppose there exist a* ρ_1-approximation *algorithm* ($\rho_1 \geq 1$) *for* $\mathcal{P}_1$ *and a* ρ_2-relaxed *algorithm* ($\rho_2 \geq 1$) *for* $\mathcal{P}_2$. *Then there exists a* ($\rho_1\rho_2$)-approximation *algorithm for* $\mathcal{P}_0$.

The detailed proof can be found in Appendix A.2. Inspired by Theorem 2, in order to solve $\mathcal{P}_0$ with performance guarantee, we only need to design approximation algorithms for the subproblems $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively.

3 SCHEDULING FLOWS IN BATCH

In this section, we design a hybrid-routing solution in reconfigurable DCNs, *WiRo*, to address the challenges of $\mathcal{P}_0$. Based on the decomposition of the original hybrid routing problem $\mathcal{P}_0$, we design two algorithms, *WiLS* and *WiRS*, to solve the two subproblems $\mathcal{P}_1$ and $\mathcal{P}_2$, with a proven guaranteed approximation ratio for each. Finally, the central controller will output the routing paths for large flows and an IS to guide the setup of wireless links.

3.1 Approximation algorithm for $\mathcal{P}_1$

We denote $\mathbf{x}_I$ as the *incidence vector* of an IS I in a conflict graph $G_c(V_c, E_c)$. $\mathbf{x}_I$ is a vector containing $|V_c|$ elements, where its j-th element is 1 if and only if the vertex $v_j \in V_c$ is an element of I, otherwise it is 0. The *independence-set polytope* P of G_c is defined as the convex

Table 1. Problems and Descriptions

Problem	Description
$\mathcal{P}_0$	The original hybrid routing problem
$\mathcal{P}_1$	Relaxing $\mathcal{P}_0$ from $x_{ij}^k \in \{0,1\}$ to $x_{ij}^k \in [0,1]$
$\mathcal{P}_2$	$\mathcal{P}_0$ without wireless interference constraints (1)(2)(3)
$\widetilde{\mathcal{P}}_2$	Relaxing $\mathcal{P}_2$ from $x_{ij}^k \in \{0,1\}$ to $x_{ij}^k \in [0,1]$
$\mathcal{P}_3$	Relaxing $\mathcal{P}_1$ from one IS selection to multiple ISs
$\widetilde{\mathcal{P}}_3$	μ-approximation of $\mathcal{P}_3$ by replacing P with Q'

Algorithm 1 WiLS: **Wi**reless **L**ink **S**etup

1: Relax the selection of optimal integer point $\mathbf{x}_I$ to the selection of optimal fractional point $\mathbf{x}' \in P$.
2: Approximate the polytope P by a polynomial-representable polytope Q' to get problem $\widetilde{\mathcal{P}}_3$.
3: Get the optimal fractional point $\mathbf{x}'$ by solving $\widetilde{\mathcal{P}}_3$.
4: Get the decomposed ISs and select one IS.
5: Fix the selected IS and solve the remaining LP problem to get $\{x_{ij}^k\}$.

hull of the incidence vectors of ISs in G_c. For any IS in G, we can find its corresponding optimal objective for $\mathcal{P}_1$ in polynomial time by fixing the IS and solving the remaining LP problem. Hence solving $\mathcal{P}_1$ is equivalent to finding the optimal incidence vector $\mathbf{x}_I$ in the polytope P which achieves the optimal objective. Since there are exponentially many incidence vectors in the polytope P, it is hard to find the optimal one.

To address this issue, we design an approximation algorithm WiLS (**Wi**reless **L**ink **S**etup, **Algorithm 1**), which relaxes the selection of the optimal integer point $\mathbf{x}_I$ to the optimal fractional point $\mathbf{x}'$ in the polytope P, i.e., $\mathbf{x}' \in P$ is a convex combination of the incidence vectors in the polytope P (Section 3.1.1). Then WiLS approximates the polytope P by another polynomial-representable polytope $Q' \subseteq P$ (Section 3.1.2). Next, WiLS decomposes the fractional point $\mathbf{x}' \in Q'$ into a polynomial number of incidence vectors which forms the convex combination. Finally, WiLS *carefully* selects one incidence vector to be the final output with a guaranteed approximation ratio (Section 3.1.3).

3.1.1 **Relaxation of IS selection**. Denote $\mathcal{I}$ as all the ISs $\{I_1, I_2, ..., I_K\}$ in G. First, WiLS relaxes the selection of only one incidence vector $\mathbf{x}_I$ to the selection of a convex combination of incidence vectors of multiple ISs, i.e., find a fractional point $\mathbf{x}' = \sum_{I\in\mathcal{I}} p_I \mathbf{x}_I$, where $p_I \geq 0$ and $\sum_{I\in\mathcal{I}} p_I \leq 1$.

An IS can be considered as the wireless links that can be scheduled simultaneously. The entire scheduling period is divided into multiple segments. For an IS $I_i \in \mathcal{I}$, we use $y(I_i)$ to denote the fraction of time allocated to it, i.e., during which the links in I_i are active simultaneously. We have the relaxation of $\mathcal{P}_1$ as a new problem $\mathcal{P}_3$:

$$\min \ \lambda$$

s.t.

$$\sum\nolimits_{f_k\in F} d^k x_{ij}^k \leq \lambda \mathcal{C}_{ij}, \ \forall e_{ij} \in E_w \tag{9}$$

$$\sum\nolimits_{f_k\in F} d^k x_{ij}^k \leq \lambda \mathcal{C}_{ij} \sum\nolimits_{e_{ij}\in I} y(I), \ \forall e_{ij} \in \overline{E}_s \tag{10}$$

$$\sum\nolimits_{I\in\mathcal{I}} y(I) \leq 1, \ \forall I \in \mathcal{I} \tag{11}$$

$$y(I) \geq 0, \ \forall I \in \mathcal{I} \tag{12}$$

$$0 \leq x_{ij}^k \leq 1, \ \forall e_{ij} \in E_w \cup \overline{E}_s, \ \forall f_k \in F \tag{13}$$

Constraints (5)(6)(7)

where the original wireless interference constraints (1)(2)(3) for only one IS selection in $\mathcal{P}_1$ are replaced by new constraints (10)(11)(12) for supporting multiple ISs in $\mathcal{P}_3$.

3.1.2 **Approximate independence-set polytope**. Since the formulation of the independence-set polytope P in problem $\mathcal{P}_3$ has an exponential number of variables $\{y(I) : I \in \mathcal{I}\}$, a direct LP solution would run in exponential time with respect to the number of variables.

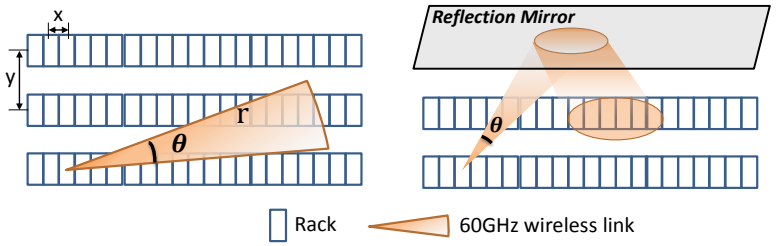


Fig. 3. 2D interference vs. 3D interference

Instead, we will exploit the 60GHz wireless features in DCN to approximate the problem $\mathcal{P}_3$ within a tight factor with only a polynomial number of variables. We first present how to approximate the polytope P by another polynomial-representable polytope Q within a guaranteed factor ρ.

General polytope approximation. The general approximation of an independence-set polytope is a well-researched area [30, 45]. We first introduce the basic notations. Consider a conflict graph $G_c(V_c, E_c)$ with n vertices and a vertex ordering $\sigma = \langle v_1, v_2, ..., v_n \rangle$ of V_c. Denote the weight of a vertex v_i as $w(v_i)$, and $N(v_i)$ as the *neighbors* of v_i, i.e, the vertices in V_c that are adjacent to v_i. Denote $\Gamma_\sigma(v_i)$ as the *backward neighbors* by the ordering σ, i.e., the vertex set of all the neighbors of v_i in $\{v_1, v_2, ..., v_{i-1}\}$. Denote $w \in R_+^n$ as an n-sized positive vector $\{w(v_1), w(v_2), ..., w(v_n)\}$. Based on these definitions, we define an independence-set polytope:

$$Q = \{w \in R_+^n : \max_{1 \le i \le n} \{w(v_i) + \sum_{v_j \in \Gamma_\sigma(v_i)} w(v_j)\} \le 1\}$$

It is proven that there exists a constant ρ that makes $Q \subseteq P \subseteq \rho Q$ hold [45]. For simplicity, we denote Q as the ρ-approximation polytope of P and ρ as the approximation ratio of general polytope approximation.

For a vertex set $X \subseteq V_c$, the subgraph of G_c induced by X is denoted by $G_c[X]$. The approximation ratio ρ is defined to be the maximum size of any IS of $G_c[v_i \cup \Gamma_\sigma(v_i)]$ for $1 \le i \le n$.

Although the above approximation works well with a constant ratio ρ in an arbitrary conflict graph, the ratio ρ may become large for a specific wireless network, such as the one based on 60GHz wireless in DCN. Existing work on the optimization of ρ in wireless environment is performed based on the circular interference model of omnidirectional antenna (e.g., $\rho \le 23$ for 802.11 wireless networks [45]). However, for 60GHz transmissions in DCNs, its rotation flexibility and small interference angle can lead to a much larger ρ than that in the conventional circular interference models if an arbitrary graph of the wireless network is allowed.

Optimization of 60GHz wireless in DCN. In the following, we will show that it is possible to utilize the 60GHz wireless inference model and topology features of data centers to obtain a lower approximation ratio. As Fig. 3a shows, the fan-shape interference range of the 60GHz link can be defined as $\mathcal{F}(\theta, r)$, which is a sector with an angle θ and a radius r. Denote $\mathcal{G}(x, y)$ as the grid topology of the data center where the row interval of racks is x and the column interval of racks is y. Denote $J(\mathcal{F}(\theta, r), \mathcal{G})$ as the maximum number of racks that are located in the range of any sector $\mathcal{F}(\theta, r)$ with its center at a rack in $\mathcal{G}$.

Based on the above definitions, we have the following lemma for ρ value when applying the 60GHz wireless in DCN:

LEMMA 1. *Given the data center topology $\mathcal{G}$ and the 60GHz wireless inference model $\mathcal{F}$, the approximation ratio ρ of the general polytope approximation is at most $J(\mathcal{F}(2\pi, r), \mathcal{G})$, where $J(\mathcal{F}(2\pi, r), \mathcal{G}) \leq \frac{2\pi}{\theta} J(\mathcal{F}(\theta, r), \mathcal{G})$.*

See the detailed proof in Appendix A.3. The above bound $J(\mathcal{F}(2\pi, r), \mathcal{G})$ can be achieved for ρ when θ is small. Since the 60GHz antenna is highly directional with a small θ and the rack placement in $\mathcal{G}$ is also dense in today's data centers [52, 53], the resulted approximation factor ρ would be large for a data center.

Next, we show how to utilize the interference features of the directional antenna to obtain an approximation polytope Q' for P with a lower approximation ratio. First, we define an edge direction ℓ by considering the asymmetry of the fan-shape interference range of the 60GHz link as follows: for any two conflict links e_{ij} and e_{uv}, if v_j is within the interference range of v_u and v_v is within the interference range of v_i, take an arbitrary edge direction between e_{ij} and e_{uw} in the conflict graph G_c; if v_j is within the interference range of v_u, take the edge direction from e_{ij} to e_{uv}; otherwise, take the edge direction from e_{uv} to e_{ij}.

By forcing the edge direction ℓ on the undirected conflict graph G_c, the neighbours of each vertex v_i are divided into the in-neighbour set $N_\ell^+(v_i)$ and the out-neighbour set $N_\ell^-(v_i)$. Then we define a new polytope:

$$Q' = \{w \in R_+^n : \max_{1 \leq i \leq n} \{w(v_i) + 2 \sum_{v_j \in N_\ell^+(v_i)} w(v_j)\} \leq 1\}$$

Now we show that Q' approximates the original polytope P within a tighter approximation ratio μ:

THEOREM 3. *Given the edge direction ℓ based on the directional 60GHz link, Q' is a μ-approximation polytope of P (i.e., $Q' \subseteq P \subseteq \mu Q'$). Let ρ^+ denote the maximum size of any IS in $G_c[N_\ell^+(v_i)]$ for $1 \leq i \leq n$. The approximation ratio $\mu = \max\{1, 2\rho^+\}$, where $\rho^+ \leq J(\mathcal{F}(\theta, r), \mathcal{G})$.*

See the detailed proof in Appendix A.4. Since the small antenna angle θ and short transmission distance r of 60GHz link, $J(\mathcal{F}(\theta, r), \mathcal{G})$ is bounded by a small constant independent of the network size [25]. It contributes to a much smaller factor μ and thus tighter approximation compared to the general factor ρ.

Based on Theorem 3, if we replace the polytope P described in $\mathcal{P}_3$ by the new polytope Q' to generate $\widetilde{\mathcal{P}}_3$, the resulted LP problem $\widetilde{\mathcal{P}}_3$ is the μ-approximation of $\mathcal{P}_3$. Specifically, denote the weight of a vertex $v_e \in V$ (v_e corresponds to a wireless link $e_{ij} \in \overline{E}_s$) as $w(v_e) = w(e_{ij}) = \sum_{f_k \in F} \frac{d^k x_{ij}^k}{\lambda \mathcal{C}_{ij}}$. By replacing the constraints (10)(11)(12) in $\mathcal{P}_3$ with the constraint $w \in Q'$, i.e.,

$$w(e_{ij}) + 2 \sum_{e_{uv} \in N_\ell^+(e_{ij})} w(e_{uv}) \leq 1, \ \forall e_{ij} \in \overline{E}_s \tag{14}$$

we have the μ-approximation LP problem $\widetilde{\mathcal{P}}_3$ for $\mathcal{P}_3$.

Extended optimization for 3D beamforming of 60GHz wireless in DCN. For the highly directional 60GHz antennas, besides the fan-shaped interference in the conventional 2D setting [15, 25], the *3D beamforming* technology is proposed in DCNs recently [16, 52, 53], where the 60GHz links are reflected by a large flat mirror attached to the ceiling of the data

center room. As Fig. 3b shows, the interference model is a small ellipse around the receiver. Normally, the size of the interference range in 3D case (i.e., the ellipse in Fig. 3b) is smaller than that in 2D case (i.e., the sector in Fig. 3a). Let $\widetilde{\mathcal{F}}(a, b)$ denote the ellipse interference range of the 60GHz link, where a and b are the minor axis and major axis of the ellipse respectively. Let $\widetilde{J}$ denote the maximum number of the racks in DCN that are located within the ellipse interference range $\widetilde{\mathcal{F}}$, then $\widetilde{J}$ is bounded by a small constant independent of the network size [52, 53]. We have the following corollary by the same technique in Theorem 3.

Corollary 1. *Given the edge direction ℓ based on the 60GHz link in 3D case, Q' is a μ-approximation polytope of P, where the factor μ is at most $2\widetilde{J}$.*

An interesting finding is that by using the general approximation with ρ factor, the ρ value would be even larger in the 3D case than that in 2D case. In the 3D space, more flexible concurrent wireless links can interfere the transmission of a single wireless link, which directly increases the ρ value based on its definition. On the contrary, using our optimization with factor μ, the μ value would become even smaller in the 3D case. Since the factor μ only includes one interference direction, the transmission of a wireless link in the 3D case will interfere much fewer concurrent wireless links at other racks, which further decreases the μ value based on its definition. For a conventional data center network with the physical layout in [52] and realistic wireless setting based on the experimental studies over our 60GHz testbed (Section 5.1), we find that ρ is larger than 80, while μ is less than 20 for the 2D case and less than 10 for the 3D case, i.e., the approximation ratio is reduced by 75% and 87% respectively. This means that our scheme performs much better than the previous schemes in the worst case.

3.1.3 **Convex decomposition and selection of IS**. After getting the optimal fractional point $\mathbf{x}'$ by solving $\widetilde{\mathcal{P}}_3$, we use the first-fit fractional weighted coloring (F^3WC) algorithm in [45] to perform the convex combination decomposition[3]. Consider a color variable in F^3WC as an IS in our problem context, it is equivalent to giving the decomposed ISs $\mathcal{I}$ in $\mathbf{x}'$ and the fraction time $\{y(I) : I \in \mathcal{I}\}$, then we have

$$\sum_{I\in\mathcal{I}} y(I) \leq \max_{1\leq i\leq n} \{w(v_i) + 2 \sum_{v_j\in N_\ell^+(v_i)} w(v_j)\} \leq 1 \tag{15}$$

Hence it ensures that we get the required limited number of decomposed ISs in polynomial time. We then select one IS I^* with a probability of its combination coefficient $y(I^*)$. When $\sum_{I\in\mathcal{I}}\{y(I)\}$ is less than 1, an *empty* IS is selected with a probability $1 - \sum_{I\in\mathcal{I}}\{y(I)\}$. Finally, we fix the selected IS in $\mathcal{P}_1$ and solve the remaining LP problem. To show the basic performance of Algorithm 1, we have the following lemma:

Lemma 2. *The expectation value of the solution obtained by Algorithm 1 is at most μ times the optimal solution of the subproblem $\mathcal{P}_1$.*

See the detailed proof in Appendix A.5. Let n denote the number of nodes in the hybrid network. Based on the above lemma, we give the following theorem to ensure the performance guarantee of Algorithm 1.

Theorem 4. *Algorithm 1 achieves an $O(\frac{\mu \log n}{\log \log n})$ approximation solution for the subproblem $\mathcal{P}_1$ with high probability.*

[3]Note that other methods of convex combination decomposition can also be applied here in our solution, such as the linear programming method in [13].

See the detailed proof in Appendix A.6. As the probability that the algorithm returns an $O(\frac{\mu \log n}{\log \log n})$ approximation solution for $\mathcal{P}_1$ is $1 - \frac{1}{n}$, which is very close to 1 when the number of nodes n is very large, we express it concisely by saying that the algorithm achieves an $O(\frac{\mu \log n}{\log \log n})$ approximation solution with high probability.

3.2 Approximation algorithm for $\mathcal{P}_2$

After solving $\mathcal{P}_1$ by Algorithm 1, we get the approximation IS $\{\widetilde{e}_{ij}\}$ and the fractional flows $\{\widetilde{x}_{ij}^k\}$. Consider $\mathcal{P}_2$ with the IS $\{\widetilde{e}_{ij}\}$ as its input $\overline{E}_s$, then this solution is also the optimal solution for the LP relaxation of $\mathcal{P}_2$. In the following, we round this fractional solution to construct the feasible integer solution for $\mathcal{P}_0$ with performance guarantee.

Algorithm 2 WiRS: **Wi**reless **R**oute **S**cheduler

1: Solve subproblem $\mathcal{P}_1$ by Algorithm 1
2: Decompose the flows $\{x_{ij}^k\}$ to path sets $\{P^k : f_k \in F\}$ and each path p is attached with a demand $\mathcal{D}(p)$
3: **for** $f_k \in F$ **do**
4: Select exactly one path p^* with probability $\frac{D(p^*)}{d^k}$
5: $p^k \leftarrow p^*$
6: **end for**
7: **return** $\{p^k\}$, $\{\gamma_{ij}\}$

We first decompose the fractional flow $\{\widetilde{x}_{ij}^k\}$ to multiple routing paths. The details of the solution for subproblem $\mathcal{P}_2$ are presented by lines 2-6 in **Algorithm 2** (called *WiRS*). The output is the set of paths P^k and each routing path $p \in P^k$ is attached with a fractional flow demand $\mathcal{D}(p)$. For each flow $f_k \in F$, we independently select the path $p \in P^k$ with a probability $\frac{\mathcal{D}(p)}{d^k}$. Based on the randomized rounding theory in [38], lines 2-6 in Algorithm 2 achieve an $O(\frac{\log n}{\log \log n})$-approximation solution for the subproblem $\mathcal{P}_2$ with high probability. In our context, it is also an $O(\frac{\log n}{\log \log n})$-*relaxed* algorithm for $\mathcal{P}_2$.

Combined with Theorem 2 and Theorem 4, WiRS will generate a solution that is less than $O(\mu(\frac{\log n}{\log \log n})^2)$ times the optimal solution of $\mathcal{P}_0$ with a probability close to 1. Thus we have the following theorem:

THEOREM 5. *Algorithm 2 is an $O(\mu(\frac{\log n}{\log \log n})^2)$ approximation algorithm for $\mathcal{P}_0$ with high probability.*

See the detailed proof in Appendix A.7. Suppose the network has m edges and there are k large flows to handle in a scheduling period. Since the LP problems solved in WiRS are splittable (fractional) multicommodity flow problems, there exists an almost-linear-time $O(m^{1+o(1)}\varepsilon^{-2}k^2)$ algorithm to produce a $(1-\varepsilon)$ approximation solution for them [35]. Other procedures in WiRS take $O(m(k+1))$ operations. Hence the total time complexity of WiRS can be optimized to $O(m^{1+o(1)}\varepsilon^{-2}k^2)$ with an additional $1-\varepsilon$ approximation factor, which achieves an almost-linear scalability with network scale in terms of the edge number m.

As a remark, WiRS can also be generalized to solve other interference models for the reason that the approximation ratio ρ in approximating the problem $\mathcal{P}_1$ is bounded for many other wireless interference models. For example, for the protocol interference model (PIM) of omnidirectional antennas, the ratio ρ is shown to be bounded by a small constant [45]. Besides, for the SINR-based physical interference model (PHY), the ratio ρ is also shown to

be bounded by $log(n)$ in [29], i.e., the approximation ratio of our solution in PHY model is $O(\log^3(n)/\log\log^2 n)$. However, it requires the controller to deal with link capacity variation based on the real-time transmissions of neighboring links.

4 SCHEDULING FLOWS IN SEQUENCE

As WiRo takes a long time to solve complex LP problems and achieve the performance guarantee, it is better fit for data centers with long lasting flows. Data center operators can run WiRo periodically based on traffic prediction according to their traffic pattern [17]. To reduce the computational complexity and support the operations of data centers with highly dynamic traffic that cannot be easily predicted, we further design an *online flow scheduler* to run with low complexity.

The online flow scheduler (OFS) consists of a *fast link setter* (FLS) and a *fast route scheduler* (FRS). It handles arrived flows in real-time. To achieve high network performance and avoid packet loss or reordering caused by network update and flow rerouting, we jointly consider network load balancing and reconfiguration cost in OFS.

4.1 Fast link setter

We design a fast link setter (FLS) to reconfigure network topology according to flow demands. We show the pseudo-code of FLS in **Algorithm 3**. In the case that one or several large flows arrive simultaneously, FLS first aggregates the flows that come from the same source rack and are destined to the same destination rack, and handle them as one flow. We denote the set of arrival flows as F_a and its size as $|F_a|$. In addition, we denote the set of wireless links in $\overline{E}_s \backslash E_s$ that do not conflict with links in E_s as E_a ($E_a \cup E_s \subseteq \overline{E}_s$). When wireless links in E_a are activated or wireless links in E_s are deactivated, E_a and E_s will be updated immediately.

Algorithm 3 FLS: **F**ast **L**ink **S**etter

```
1: Sort flows in F_a in the descending order of size
2: for each flow in F_a do
3:    Find the K-shortest paths: {P_1, P_2, ..., P_K}
4:    for i = 1 to K do
5:       if P_i can be built then
6:          Add wireless links in P_i ∩ E_a to E_s
7:          Update E_a and E_s
8:          Break
9:       end if
10:   end for
11: end for
12: return E_a, E_s
```

In Algorithm 3, on line 1, we sort all the flows in F_a in the descending order of size. On lines 2 to 11, we build wireless links one by one, and update E_a and E_s accordingly. For each flow $f_k \in F_a$, we search its K-shortest paths ($K \geq 1$) without considering the interference between wireless links. We then traverse these K-shortest paths, and determine whether each of them can be built. After we find one available routing path, we stop the traversal and update E_s and E_a. Since the K-shortest paths of a flow can be found using Yen's algorithm [50], whose time complexity is $O(K \cdot |V| \cdot (|E| + \log|V|))$, the time complexity of

FLS is $O(|F_a| \cdot (K \cdot |V| \cdot (|E| + \log |V|))$. We conduct evaluations in Section 5.3 to see the practical impacts of different K on the flow performance.

4.2 Reconfiguration cost

In reconfigurable DCNs, the wireless links are flexible and can be built or removed flexibly. If there is a need to reroute the flows in the middle of transmission, i.e., in-transit flows, to other paths for optimal traffic engineering, the affected flows will experience different levels of performance degradation. Some example performance impacts are the packet loss and retransmission. Different applications have different sensitivities to packet loss and retransmission. Rerouting will also incur extra cost for scheduling and route management. If a flow has very few unsent data packets left, the relative cost is higher. To avoid rerouting flows that are almost finished and prevent frequent rerouting, we introduce a reconfiguration cost β as

$$\beta = \sum\nolimits_{f_k \in F_c} (1 - \frac{\widetilde{d}^k}{d^k})$$

where F_c denotes the set of in-transit flows whose routing paths change, d^k denotes the total size of f_k and $\widetilde{d}^k$ denotes the size of unsent data of f_k by the current time slot. Correspondingly, we denote the set of all in-transit flows as F_s. It is obvious that $F_c \subseteq F_s$.

In addition to the definition of reconfiguration cost, we define the set of links that f_k flows through as R_k. Then we have the following equality if the routing path of f_k has not been changed during the topology reconfiguration:

$$x_{ij}^k = 1, \ \forall e_{ij} \in R_k \tag{16}$$

Combined with (16), we have the following claim:

Claim 1. *For a flow $f_k \in F_s$ that has gone through reconfiguration of the data center network, if constraints (5)(6)(7) and (16) are satisfied at the same time, its routing path stays the same.*

See the detailed proof in Appendix A.8. We define a binary variable c^k to indicate whether the routing path of $f_k \in F_s$ has been changed during the reconfiguration or not, and c^k is represented as:

$$c^k = \begin{cases} 1, & \exists \ x_{ij}^k = 0 \text{ for all } e_{ij} \in R_k \\ 0, & \forall \ x_{ij}^k = 1 \text{ for all } e_{ij} \in R_k \end{cases} \tag{17}$$

Equivalently,

$$c^k = \max_{e_{ij} \in R_k} \{1 - x_{ij}^k\} \tag{18}$$

Thus, the reconfiguration cost can be represented as follows:

$$\beta = \sum\nolimits_{f_k \in F_s} c^k \cdot (1 - \frac{\widetilde{d}^k}{d^k})$$

In the following, we will introduce how to schedule flows by taking into account the reconfiguration cost.

4.3 Fast route scheduler

We propose FRS to reschedule the in-transit flows and find routing paths for arrival flows, with the goal of balancing the network load and reconfiguration cost. Since the wireless links in E_s have been determined with FLS, it is feasible for us to design the routing with static topology.

Algorithm 4 FRS: **F**ast **R**oute **S**cheduler

1: Set wireless links by Algorithm 3
2: Calculate the optimal fractional point $\widetilde{x}_{ij}^k$ by solving $\widetilde{\mathcal{P}}_4$
3: Decompose $\{\widetilde{x}_{ij}^k\}$ to path sets $\{P^k{:}f_k \in F_s \cup F_a\}$
4: **for** $f_k \in F_s \cup F_a$ **do**
5: Select exactly one path p^* with probability $\frac{\mathcal{D}(p^*)}{\widetilde{d}^k}$
6: $p^k \leftarrow p^*$
7: **end for**
8: **return** $\{p^k\}$, $\{\gamma_{ij}\}$

Although we can obtain the minimum reconfiguration cost by changing no routing paths of flows in F_s, when new flows arrive, it is probably not a global optimal solution. Since we also need to control the performance loss caused by the topology reconfiguration, we add the reconfiguration cost into $\mathcal{P}_2$ (as shown in Table 1). Then, the optimization objective can be expressed as:

$$\Phi = \lambda + w \cdot \beta$$

where $w \cdot \beta$ is the weighted reconfiguration cost with the weight w set by data center operators.

We have two approaches to solve the above problem. In the first approach, we first ensure that all routing paths of flows in F_s remain unchanged, thus the reconfiguration cost is 0. Based on this, we can use WiRS to choose routing paths for flows in F_a. We denote the optimal objective value as obj_1, in the case that no routing paths of in-transit flows have been changed. The second one is to minimize the objective value directly, which considers load balancing and reconfiguration cost simultaneously. We formulate the problem $\mathcal{P}_4$ as:

$$\min \ \Phi$$

s.t.

$$1 - x_{ij}^k \le c^k, \ \forall f_k \in F_s, \ e_{ij} \in R_k \tag{19}$$

$$x_{ij}^k = 0 \text{ or } 1, \ \forall f_k \in F_s \cup F_a, \ \forall e_{ij} \in E \tag{20}$$

Constraints (4) − (7)

We relax the value range of x_{ij}^k in (20) to $[0, 1]$, and get the relaxed version of $\mathcal{P}_4$, which we denote as $\widetilde{\mathcal{P}}_4$. We denote its optimal objective value as obj_2. Although the first approach has a lower complexity than the second one, it leads to a higher objective value than that of the second one. Since the solution corresponding to the first approach is a feasible solution of $\widetilde{\mathcal{P}}_4$, and obj_2 is the optimal (minimum) objective value of $\widetilde{\mathcal{P}}_4$, we have $obj_1 \ge obj_2$.

Based on the above analysis, we design a fast route scheduler (**FRS, Algorithm 4**) on the basis of the second approach. The algorithm constructs the feasible integer solution for $\mathcal{P}_4$ with a performance guarantee. In Algorithm 4, we first set wireless links by Algorithm 3. Then we find the optimal solution of $\widetilde{\mathcal{P}}_4$ and decompose the fractional flows $\{\widetilde{x}_{ij}^k\}$ to multiple routing paths. The output is the set of paths P^k and each routing path $p \in P^k$ is attached with a fractional flow demand $\mathcal{D}(p)$. At last, for each flow $f_k \in F_s \cup F_a$, we independently select the path $p \in P^k$ with a probability $\frac{\mathcal{D}(p)}{\widetilde{d}^k}$. Let n denote the number of nodes in the hybrid network. We have the following theorem:

THEOREM 6. *FRS is an* $O(\frac{\log n}{\log \log n})$ *approximation algorithm for* $\mathcal{P}_4$ *with high probability.*

(a) point-to-point communication (b) Reflection-based communication

Fig. 4. Parameter measurement of 60GHz wireless

See the detailed proof in Appendix A.9. With Theorem 6, our algorithm outputs a routing solution that balances the network load and reconfiguration cost simultaneously with a proved guarantee.

5 EVALUATION

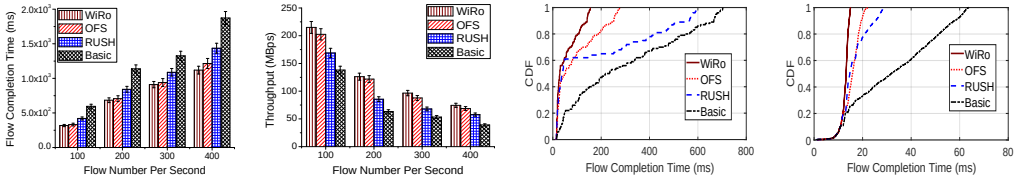
In this section, we first conduct parameter measurement in realistic 60GHz platform. Then we compare the performance of our solutions with that of existing scheduling solutions.

5.1 Parameter measurement

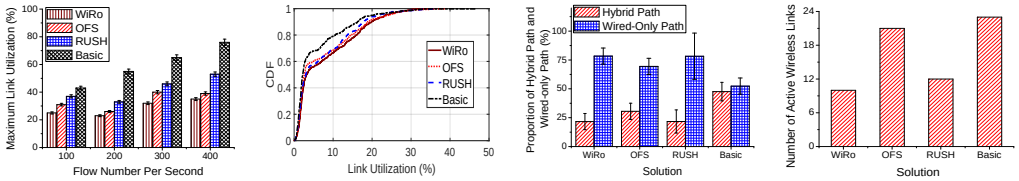
To guide our later simulations under realistic wireless settings, we measure the 60GHz parameters of interest in a realistic platform. A 60GHz link is formed with a pair of horn antennas, running over 60GHz rectangular waveguide along with a self-designed power amplifier (30dB gain and 0.37W saturated output power). The distance between the antenna transmitter and receiver is set to 10 meters. For the 2D case (Fig. 4a), we test the received signal when fixing the receiver and rotating the transmitter antenna. As Table 2 shows, due to the highly directional feature of 60GHz antenna, the communication quality decreases when the deflection angle of the transmitter antenna increases, and the main-lobe width of the 60GHz antenna is measured as 20^o. We then measure the reflection performance in the 3D case by adding one flat metal mirror between the transmitter and the receiver. By tuning the antenna angle, the signal from the transmitter can arrive at the receiver through a reflection on the mirror (Fig. 4b). We found that the impact of the reflection mirror is small and the measured bandwidth with the same distance is kept with little change, which confirms the previous measurement results in [16, 52] and indicates that the same transmission distance can be achieved for both 2D and 3D cases. The standard derivation (RSS-Std) of the received signal in Table 2 is derived by monitoring the received signal with one measured data point per second over a duration of 200 seconds. We can see that the measured signal keeps stable over time, and the measured 60GHz wireless bandwidth is between 2.5Gbps and 6Gbps in all the measured cases. It confirms the feasibility and stability to apply 60GHz wireless links for multiple-gigabit transmissions in DCN.

Table 2. Measurement result of 60GHz antenna

	0^o deflection	10^o deflection	20^o deflection
RSS-Avg (dbm)	-19	-32	-38
RSS-Std (dbm)	0.29	0.51	0.81
Bit Error Rate	0	0	1.325e-3



(a) Completion time of large flows (b) Throughput of large flows (c) Completition time of large flows (FN=100) (d) Completition time of small flows (FN=100)



(e) Maximum link utilization (f) Utilization of wired links (FN=100) (g) Proportion of different routing paths (FN=100) (h) Number of active wireless links (FN=100)

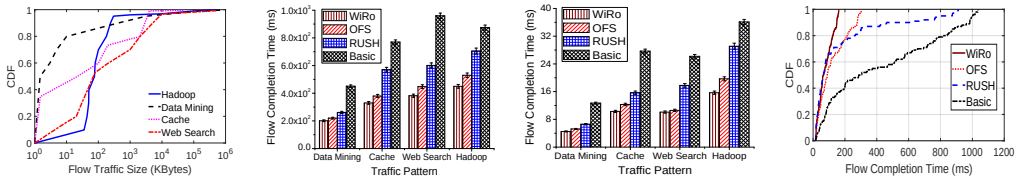
Fig. 5. Performance of solutions with different flow arrival rates

5.2 Simulation setup and methodologies

We implement an iterative flow-level simulator for flow scheduling and routing in C++, which calls the Gurobi Optimizer [1] to solve the LP problems. We use the same TCP flow setting as that in [3], and the basic data center topology that consists of 32 racks is formed with a fat-tree [2], with the configuration of the racks as the physical layout in [52]. Racks are grouped into 2×2 clusters, each consisting of a row of 8 racks with no inter-spacing. Clusters are separated with aisles, at 3m between columns and 2.4m between rows, respectively. The wireless transmission follows the general physical interference and path loss model [15], and the relevant wireless parameters such as the bandwidth and antenna angle are all set following the testbed-based measurements in Fig. 4. The Rayleigh fading model is applied to simulate the dynamics in wireless channels and the actual transmission rate is based on the channel conditions.

For the comparative analysis, we evaluate the performance of our solutions (WiRo and OFS) and existing flow scheduling algorithms (RUSH [27], Flyway [25], GFF [3] and ECMP [24]) used in other traffic engineering proposals in DCNs. RUSH is a wireless scheduling and routing algorithm with a proved approximation ratio [27], but it considers an ideal wireless model without interference between any wireless link pairs. Flyway uses a greedy scheme based on the flow demands to build the wireless links [25], and then solves the remaining LP-based routing problem. GFF is a routing algorithm using the Global First Fit heuristic to schedule flows in DCNs [3]. ECMP is the most popular random routing strategy applied in DCNs [31].

In our studies, we take Flyway+ECMP as the baseline algorithm for comparison, denoted as "Basic". Since GFF and ECMP are originally designed for wired DCNs only, to make a fair comparison, we apply the same methods to configure additional wireless links for all of them. If there are no special instructions for OFS, we set the trade-off parameter w to 0 and the number of shortest paths searched by FLS to 5. We use the flow traces from the public data of real data center traffic posted by [9] as our system input, and take the flows larger than 100KBytes as large flows.



(a) Flow size of different applications (b) Completion time of large flows (c) Completion time of small flows (d) Completion time of large flows (Data Mining)

Fig. 6. Performance of solutions with different traffic patterns

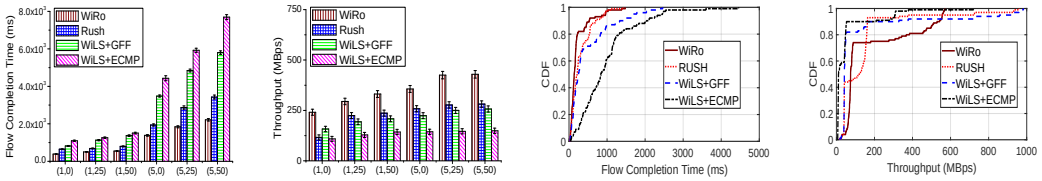
5.3 Comparative analysis

We conduct evaluations to compare different solutions and present the results in following.

5.3.1 Performance with different solutions. We compare WiRo and OFS with Basic and RUSH under different flow arrival rates (i.e., flow number per second, FN) and show the results in Fig. 5. As Fig. 5a shows, OFS and WiRo perform better than RUSH and Basic on the completion time of large flows. The average completion time of large flows in WiRo and OFS is up to 72.6% and 41.8% lower than that of Basic. WiRo and OFS reduce the flow completion time of RUSH by 65% and 24.8% on average. In Fig. 5b, the throughput of WiRo is up to 99% and 47% higher than that of Basic and RUSH. Benefited from the good wireless setups and routing strategies, both WiRo and OFS can outperform RUSH and Basic on flow completion time and throughput.

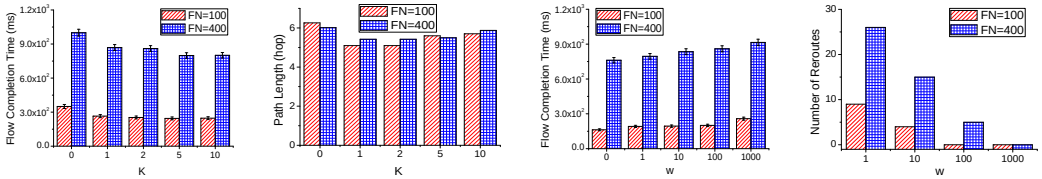
As shown in Fig. 5c, when FN is 100, more than 60% large flows in WiRo, OFS and RUSH have a completion time less than 100ms. The maximum flow completion time is 173ms and 241ms for WiRo and OFS, which is much less than that of RUSH and Basic (600ms and 719ms). This is because WiRo and OFS try to balance the network load by carefully building wireless links and adaptively reconfiguring hybrid topology according to traffic demands. In Fig. 5d, the completion time of more than 95% small flows in WiRo and OFS is less than 20ms, while only 80% and 35% of small flows in RUSH and Basic have a completion time less than 20ms. This is because WiRo and OFS schedule large flows carefully to balance network load and reduce the network congestion. Benefitted from the balanced network load, WiRo does not incur very large delay.

In Fig. 5e, we compare the maximum link utilizations of the four solutions under different flow arrival rates. Obviously, the maximum link utilization increases with the network load on the whole. As WiRo adopts well-designed reconfigurable topology and routing paths by running WiLS and WiRS, it achieves the lowest maximum link utilization among these four solutions. As Basic adopts ECMP strategy to transfer flows, it prefers to allocate flows to shorter hybrid routing paths. As a result, the wired link capacities have not been fully used but the wireless links are overused. Compared with other solutions, Basic has the highest maximum link utilization. Benefitted from the well-designed topology reconfiguration and flow routing algorithms, the maximum link utilizations of our solutions (WiRo and OFS) are lower than that of RUSH and Basic, which demonstrates the advantage of our solutions on balancing the DCN load. The utilization of wired links is shown in Fig. 5f. As Basic builds wireless links greedily and schedules flows to transmit over the shortest path, many flows are transferred with wireless links, thus many wired links have a lower link utilization. The maximum link utilization of Basic is more than 45%, which is higher than that of other solutions. Since WiRo makes full use of the bandwidth of wired and wireless links, its link utilization is slightly higher than that of OFS and RUSH in general.



(a) Completion time of large flows (b) Throughput of large flows (c) Completion time of large flows (X=1, Y=50) (d) Throughput of large flow (X=1, Y=50)

Fig. 7. Performance of WiRo with different hotspots



(a) Completion time of large flows with different K (b) Avarage path length of all flows (c) Completion time of large flows with different w (d) Reroute number

Fig. 8. Performance of OFS with different parameter settings

To demonstrate the participation of wireless links in flow transferring, we classify the routing paths into two categories: *wired-only* paths formed with wired links and *hybrid* paths consisting of wired and wireless links. In Fig. 5g, we show the proportions of flows that are transferred through wired-only paths and hybrid paths in different solutions. In Basic, almost half the large flows are transferred through hybrid paths. As all flows are delivered with shortest paths and wireless links can shorten the wired-only paths, many flows are allocated to hybrid paths. In Basic, the wireless links become bottlenecks and affect the flow performance. Since OFS adopts a greedy method to set up wireless links to transfer more large flows and calls FRS to balance the network load, more than 30% flows are transferred with hybrid paths. In WiRo and RUSH, about 25% large flows are transferred on hybrid paths. In Fig. 5h, when FN is 100, the numbers of active wireless links in WiRo and RUSH are 10 and 12, respectively. They build wireless links by taking into account the whole flow matrices. OFS and Basic build wireless links greedily, and have 21 and 23 active wireless links respectively, which is higher than those of WiRo and RUSH.

5.3.2 Performance with different traffic patterns. We use four typical application traffic patterns from functional DCNs, including the workloads of data mining[24], web searching [4], Hadoop [39] and caching [39], to learn the performance of our solutions and existing solutions. We present the distribution of flow sizes over different applications in Fig. 6a.

In Fig. 6b, WiRo and OFS reduce the large flow completion time of Basic by up to 60.2% and 53.3% respectively. Their large flow completion times are up to 72% and 33.3% lower than that of RUSH. We show the small flow completion time in Fig. 6c. Benefitted from the better strategies for setting wireless links and routing large flows in our solutions, the flow completion times of small flows in WiRo and OFS are up to 64.5% and 58.7% lower than that of Basic. The results also show that our solutions can work well with different traffic patterns.

We present the distributions of flow completion time of different solutions over data mining traffic in Fig. 6d. Affected by both the greedy method for building wireless links and the random routing strategy, more than 60% large flows in Basic have a completion time exceeding 200ms, while more than 80% large flows in RUSH and OFS have a completion time less than 200ms. Benefitted from topology reconfiguration, the maximum flow completion time in OFS is 302ms, which is much less than that of RUSH. With the completion time of all flows less than 161ms, WiRo performs better than others. This is because it builds wireless links by taking into account the traffic patterns and schedules flows by taking advantages of the solutions of related LP problems.

5.3.3 Performance with hotspots. As the prevalent hotspots in data centers significantly reduce the performance of services and networks [16]. In Fig. 7, we scale up a certain percent of flow sizes to simulate heavy network load and evaluate the performance of WiRS and other scheduling schemes, RUSH, GFF and ECMP, on handling data center hotspots. To make a fair comparison, we set up the wireless links with WiLS for GFF and ECMP. We use a pair (X, Y) to denote the popular *hotspot* traffic in DCNs [9], where we scale up the original flow sizes X times to simulate various network loads and randomly add Y percentage of *hotspot flows*. As expected, both the average flow throughput and flow completion time increase with the average flow size and the percentage of hotspot flows. Under different settings, WiRo achieves the highest flow throughput and lowest completion time on average for all the traffic patterns. Compared to other schemes, its throughput is up to 190% higher and its flow completion time is up to 71% lower. By reconfiguring wireless links with WiLS and selecting routing paths for large flows with WiRS, WiRo achieves a network-wide load balancing, thus good flow performance.

To demonstrate the flow completion time and throughput of different solutions in DCNs with hotspot, we scale up 50% large flows and present the distributions of flow completion time and throughput in Fig. 7c and 7d. More than 95% large flows in WiRo can be completed in 1000ms, while only 85% and 60% large flows in GFF and ECMP are completed in 1000ms. With all schemes scheduling flows on the same hybrid topology, our routing solution, WiRS, performs better than existing solutions, GFF and ECMP. This is because WiRS selects routing paths based on the solution of LP problems while GFF and ECMP adopt random or heuristic routing strategies. In Fig. 7d, more than 20% large flows in WiRo have a throughput exceeding 200MBps, while less than 10% large flows in other solutions have a throughput of more than 200MBps. Compared with existing solutions, WiRS can achieve higher throughput and shorter completion time.

5.3.4 Performance with different parameter settings. To evaluate the performance of OFS under different parameter settings, we set different values for K and w, which represent the number of shortest paths that FLS searches and the weight of reconfiguration cost in FRS. We set the flow number per second to 100 and 400 respectively and show the results in Fig. 8. In Fig. 8a, $K = 0$ means that there is no wireless link in the network. When we gradually increase the value of K, the flow completion time decreases, but from $K = 5$, it decreases slowly. This may be because the scheduler finds a good decision within 5 shortest paths in most of the time. In Fig. 8b, except when K equals 0, the average path length increases with the value of K, as a larger K represents a larger search space for the scheduler to find proper wireless links but at the cost of longer path lengths. In our evaluation, adding wireless links to the wired topology reduces the flow completion time by up to 20%, which shows the advantages of reconfiguring the topology to improve the flow performance.

In Fig. 8c, we adjust the value of w to see the changes of flow completion time and the number of flow reroutes. When w is increased from 0 to 1000 and the flow arrival rate is 400 per second, the average completion time of large flows increases from 761 ms to 915 ms. Since w implies the significance of reconfiguration cost in our minimization problem, when w is very large, the flow scheduler will try to avoid rerouting to lower the optimization objective. According to our evaluation, when w is increased from 0 to 1000 and FN is 400, the number of rerouting decisions decreases from 26 to 0. In practice, data center operators can set proper w based on the sensitivity of their services to flow rerouting.

5.3.5 Runtime of WiRo and OFS. To measure the time needed by WiRo and OFS to schedule the transmissions, which is denoted as "runtime", we run them on a laptop, which is equipped with Intel Core i7-7500U 2.70GHz CPU, 8GB memory, 240GB SSD and runs Windows 10 64-bit version, under different network settings. We present the network settings and the runtime ranges under each setting in Table 3. WiRo needs to construct the hybrid topology and find good routing paths for large flows by solving LP problems. If there are 32 racks and each of them is equipped with a wireless radio, there will be more than 800 wireless links in $\overline{E}_s$, which will increase exponentially along with the number of racks. As it takes a long time to structure the hybrid topology and determine the routing paths for flows, WiRo is more efficient in DCNs with long lasting traffic, where data center operators can run it periodically based on the prediction of traffic.

OFS is an online flow scheduler that handles new arrival flows in real time. It adopts a greedy method to set wireless links, which runs quickly. In the network with 128 racks and 100 large flows, the total runtime of OFS on our equipment is less than 0.15 seconds, which could be even lower if it is run on high-performance servers in real data centers. Because the runtime of OFS is short, it is better fit for handling highly dynamic traffics in DCNs. For data centers mixed with long-lasting and dynamic traffics, operators can apply WiRo to handle long-lasting traffic periodically and OFS to deal with dynamic flows in real time.

Table 3. Runtime of WiRo and OFS

Configuration		Runtime (s)	
Rack Number	Flow Number	WiRo	OFS
8	10	$0.03 \sim 0.10$	< 0.001
	100	$0.50 \sim 0.90$	
32	10	$0.7 \sim 2$	$0.01 \sim 0.03$
	100	$28 \sim 40$	
128	10	$13 \sim 25$	$0.05 \sim 0.15$
	100	$81 \sim 660$	

6 DISCUSSION ON DEPLOYMENT ISSUES

Reconfigurable DCNs have attracted a lot of attention in recent years. In order to give full play to the advantages of network reconfiguration techniques, researchers propose many methods to construct reconfigurable data center networks from system level [16, 23, 36, 47]. Despite these new architectures show great potential in augmenting the network performance, efficient routing is still critical to getting the most out of their strengths. In response to this problem, we focus on designing efficient routing in reconfigurable DCNs.

Both centralized and decentralized routing are widely used in practical networks. Despite decentralized routing shows advantages in self adjusting and reacting to short-term network events, such as traffic jam and link failure, it has inherent flaws in achieving optimal flow scheduling. For example, optimizing the link weights for OSPF to given traffic is proved to be NP-hard, and even the best setting of the weights can deviate significantly from the optimal routing assignments [37]. As data center operators is more concerned about network performance and convenient operation, the development of SDN promotes the widespread application of centralized routing in data center networks.

To successfully deploy our solutions in practical data centers, there are some concerns to be well solved. First, our solutions assume that the demand of the arriving flows is known, while in many cases such information is difficult to obtain. Bai et al. [8] have designed novel mechanism, PIAS, to conduct information-agnostic flow scheduling in commodity data centers. By first running PIAS to classify flows into small and large ones, we can run our solutions to schedule large flows and obtain high flow performance. Second, OFS runs once for each arriving large flow, which may lead to high scheduling delay and large resource consumption in production data centers. As we use fixed threshold to classify small flows and large flows, and different data centers have different distributions of flow demands, we recommend using flexible thresholds to differentiate flows, thus balancing the resource consumption and the scheduling delay of our solutions.

7 RELATED WORK

There are many efforts on load balancing in conventional wired-only DCNs. ECMP [31] is a widely used flow-based load balancing strategy in data centers. Despite that it is easy to deploy and runs fast, ECMP suffers from well-known performance problems such as hash collisions and the unfitness for the asymmetric network topologies [44]. Some fine-grained mechanisms [33], [12], are proposed to handle its drawbacks, but none of them is fit for the asymmetric network topologies [44]. WCMP [51] and Presto [28] use weighting-based load balancing to deal with the asymmetry, where they add weights to switches or end-hosts. As they use static weights, they can only achieve sub-optimal solutions with dynamic traffic loads or reconfigurable networks. Different from these schemes, our algorithms are designed for reconfigurable network topologies, and they work well in asymmetric topologies.

Some recent studies propose to introduce wireless radios or optical techniques to data center networks to flexibly configure the network topology and improve the performance of flows. Specifically, emerging 60GHz radios, free-space optics and optical circuit switches attract the attentions of researchers and data center operators. c-Through [46] and OSA [14] are proposed to enable flexible optical links among all the ToR switches. xWeaver [47] exploits neural networks to reconfigure the optical circuit switches to achieve the self-defined QoS goals. RotorNet achieves the global network flexibility with the cycling of optical matchings [36]. With predefined network topologies, Flat-tree [48] is extended to change the network topology from one to another. Applying 60GHz wireless links in data center networks is first proposed in Flyway [25]. The authors of [52] use ceiling reflectors to bounce wireless signals to avoid blocking on the 2D plane. Diamond [16] rearranges the placement of racks to take full advantage of wireless links for higher network capacity. For free-space optics (FSO), Firefly adopts FSO in DCNs to transfer data [26]. ProjecToR enables direct links between all pairs of racks with FSO [23]. Although existing studies take advantages of configurable links in different ways, none of them has theoretically addressed the network load balancing problem in data centers. We propose different algorithms to efficiently transfer flows in reconfigurable DCNs with theoretical guarantees.

New network reconfiguration techniques offer high capability of improving flow performance by flexibly reconfiguring networks. In recent years, demand-aware networking on reconfigurable DCNs has been initiated and explored [5–7, 20, 21, 40]. The theory of demand-aware, self-adjusting networks is presented in [7], which is applied to minimize the network congestion and route lengths in demand-aware networks [6]. Foerster et al. characterize the algorithmic complexity of reconfigurable DCNs in [21]. They argue that it will result in non-optimal routing if network operators divide reconfigurable DCNs into reconfigurable and static parts, and route flows on either part "exclusively" by labeling flows as small or large. In [20], Fenz et al. provide several algorithms to jointly optimize topology and routing in reconfigurable DCNs. Inspired by these studies, we design non-segregated routing solutions in reconfigurable DCNs. Instead of minimizing the route lengths for flows, we set our goal as minimizing the maximum congestion level of all links, which has been shown to be a desirable feature of DCNs [25, 27]. Regardless of the methodologies and potential problems, the above architectures can benefit from our hybrid routing solutions.

8 CONCLUSION

In this paper, we propose two flow scheduling solutions, WiRo and OFS, for the network-wide load balancing in reconfigurable DCNs. Our solutions intend to take advantage of both the stable wired links and the flexible high-bandwidth wireless links for seamless and high performance transmissions of dynamic traffic in the DCNs. Jointly considering the interference and routing issues, both of our solutions provide fine-grained flow scheduling with the performance guarantee. Our solutions can be generalized and applied to different deployments of wireless technologies in DCNs. We build a 60GHz wireless platform to estimate the realistic wireless parameters for both the model analysis and simulations. The evaluation results demonstrate the effectiveness of our solutions in reducing the flow completion time and increasing the throughput for both large and small flows. As cloud services and network scale are growing fast, it is necessary to improve the flow processing speed of our solutions in the future.

ACKNOWLEDGEMENT

We thank our shepherd Stefan Schmid and the anonymous reviewers for their valuable feedbacks. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1800303 and in part by the NSFC Project under Grant 61872211. The work of Z. Yang was supported by the China Scholarship Council under Grant 201806210244. The work of X. Wang was supported by the NSF CNS under Grant 1526843. The work of M. Li was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU 11268616 and in part by the NSFC under Grant 11771365.

REFERENCES

[1] 2019. https://www.gurobi.com/.
[2] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. 2008. A Scalable, Commodity Data Center Network Architecture. In *SIGCOMM*.
[3] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. 2010. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *NSDI*.
[4] Mohammad Alizadeh, Albert Greenberg, David A Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data Center TCP (DCTCP). In *SIGCOMM*.
[5] Chen Avin, Kaushik Mondal, and Stefan Schmid. 2017. Demand-Aware Network Designs of Bounded Degree. In *DISC (LIPIcs)*.

[6] Chen Avin, Kaushik Mondal, and Stefan Schmid. 2019. Demand-Aware Network Design with Minimal Congestion and Route Lengths. In *INFOCOM*.

[7] Chen Avin and Stefan Schmid. 2019. Toward Demand-Aware Networking: A Theory for Self-Adjusting Networks. *ACM SIGCOMM Computer Communication Review* 48, 5 (2019), 31–40.

[8] Wei Bai, Li Chen, Kai Chen, Dongsu Han, Chen Tian, and Hao Wang. 2015. Information-Agnostic Flow Scheduling for Commodity Data Centers. In *USENIX NSDI*. 455–468.

[9] Theophilus Benson, Aditya Akella, and David A Maltz. 2010. Network Traffic Characteristics of Data Centers in the Wild. In *IMC*.

[10] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. http://pages.cs.wisc.edu/~tbenson/IMC10_Data. (2010).

[11] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. 2011. MicroTE: Fine Grained Traffic Engineering for Data Centers. In *CoNEXT*.

[12] Jiaxin Cao, Rui Xia, Pengkun Yang, Chuanxiong Guo, Guohan Lu, Lihua Yuan, Yixin Zheng, Haitao Wu, Yongqiang Xiong, and Dave Maltz. 2013. Per-packet Load-balanced, Low-Latency Routing for Clos-based Data Center Networks. In *CoNEXT*.

[13] Robert Carr and Santosh Vempala. 2000. Randomized Metarounding. In *STOC*.

[14] Kai Chen, Ankit Singla, Atul Singh, Kishore Ramachandran, Lei Xu, Yueping Zhang, Xitao Wen, and Yan Chen. 2014. OSA: An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility. *IEEE/ACM Transactions on Networking (ToN)* 22, 2 (2014), 498–511.

[15] Y. Cui, H. Wang, X. Cheng, D. Li, and A. Ylä-Jääski. 2013. Dynamic Scheduling for Wireless Data Center Networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 24, 12 (2013), 2365–2374.

[16] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Chao Zhu, Xiangyang Li, Liu Yang, and Ning Ge. 2016. Diamond: Nesting the Data Center Network with Wireless Rings in 3D Space. In *NSDI*.

[17] Yong Cui, Zhenjie Yang, Shihan Xiao, Xin Wang, and Shenghui Yan. 2017. Traffic-Aware Virtual Machine Migration in Topology-Adaptive DCN. *IEEE/ACM Transactions on Networking (ToN)* 25, 6 (2017), 3427–3440.

[18] Devdatt Dubhashi and Desh Ranjan. 1998. Balls and Bins: A Study in Negative Dependence. *Random Structures & Algorithms* 13, 2 (1998), 99–124.

[19] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers. In *SIGCOMM*.

[20] Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu. 2019. Efficient Non-Segregated Routing for Reconfigurable Demand-Aware Networks. In *IFIP Networking*.

[21] Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. 2018. Characterizing the Algorithmic Complexity of Reconfigurable Data Center Architectures. In *ANCS*.

[22] Michael R Garey and David S Johnson. 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. *WH Freeman San Francisco* (1979).

[23] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Houman Rastegarfar, Pierre-Alexandre Blanche, Madeleine Glick, Daniel Kilper, Janardhan Kulkarni, Gireeja Ranade, and Nikhil Devanur. 2016. ProjecToR: Agile Reconfigurable Datacenter Interconnect. In *SIGCOMM*.

[24] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta. 2009. VL2: A Scalable and Flexible Data Center Network. In *SIGCOMM*.

[25] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. 2011. Augmenting Data Center Networks with Multi-gigabit Wireless Links. In *SIGCOMM*.

[26] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R Das, Jon P Longtin, Himanshu Shah, and Ashish Tanwer. 2014. FireFly: A Reconfigurable Wireless Data Center Fabric Using Free-Space Optics. In *SIGCOMM*.

[27] Kai Han, Zhiming Hu, Jun Luo, and Liu Xiang. 2015. RUSH: RoUting and Scheduling for Hybrid Data Center Networks. In *INFOCOM*.

[28] Keqiang He, Eric Rozner, Kanak Agarwal, Wes Felter, John Carter, and Aditya Akella. 2015. Presto: Edge-based Load Balancing for Fast Datacenter Networks. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 465–478.

[29] Martin Hoefer, Thomas Kesselheim, and Berthold Vöcking. 2011. Approximation Algorithms for Secondary Spectrum Auctions. In *SPAA*.

[30] Martin Hoefer, Thomas Kesselheim, and Berthold Vöcking. 2014. Approximation Algorithms for Secondary Spectrum Auctions. *ACM Transactions on Internet Technology (TOIT)* 14, 2-3 (2014), 16.
[31] Christian E Hopps. 2000. Analysis of An Equal-Cost Multi-Path Algorithm. *RFC 2992, IETF* (2000).
[32] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. 2003. Impact of Interference on Multi-hop Wireless Network Performance. In *MOBICOM*.
[33] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. 2007. Dynamic Load Balancing without Packet Reordering. *ACM SIGCOMM Computer Communication Review* 37, 2 (2007), 51–62.
[34] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, and Ronnie Chaiken. 2009. The Nature of Data Center Traffic: Measurements & Analysis. In *IMC*.
[35] Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. 2014. An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations. In *SODA*.
[36] William M Mellette, Rob McGuinness, Arjun Roy, Alex Forencich, George Papen, Alex C Snoeren, and George Porter. 2017. RotorNet: A Scalable, Low-complexity, Optical Datacenter Network. In *SIGCOMM*.
[37] Nithin Michael and Ao Tang. 2014. Halo: Hop-by-Hop Adaptive Link-State Optimal Routing. *IEEE/ACM Transactions on Networking* 23, 6 (2014), 1862–1875.
[38] Prabhakar Raghavan and Clark D Tompson. 1987. Randomized Rounding: A Technique for Provably Good Algorithms and Algorithmic Proofs. *Combinatorica* (1987).
[39] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In *SIGCOMM*.
[40] Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, and Zvi Lotker. 2016. SplayNet: Towards Locally Self-Adjusting Networks. *IEEE/ACM Transactions on Networking (ToN)* 24, 3 (2016), 1421–1433.
[41] Siddhartha Sen, David Shue, Sunghwan Ihm, and Michael J Freedman. 2013. Scalable, Optimal Flow Routing in Datacenters via Local Link Balancing. In *CoNEXT*.
[42] Ji-Yong Shin, Emin Gn Sirer, Hakim Weatherspoon, and Darko Kirovski. 2013. On the feasibility of completely wirelesss datacenters. *IEEE/ACM Transactions on Networking (ToN)* 21, 5 (2013), 1666–1679.
[43] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, et al. 2015. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. In *SIGCOMM*.
[44] Erico Vanini, Rong Pan, Mohammad Alizadeh, Parvin Taheri, and Tom Edsall. 2017. Let It Flow: Resilient Asymmetric Load Balancing with Flowlet Switching. In *NSDI*.
[45] Peng-Jun Wan. 2009. Multiflows in Multihop Wireless Networks. In *MobiHoc*.
[46] Guohui Wang, David G Andersen, Michael Kaminsky, Konstantina Papagiannaki, TS Ng, Michael Kozuch, and Michael Ryan. 2010. c-Through: Part-time Optics in Data Centers. In *SIGCOMM*.
[47] Mowei Wang, Yong Cui, Shihan Xiao, Xin Wang, Dan Yang, Kai Chen, and Jun Zhu. 2018. Neural Network Meets DCN: Traffic-driven Topology Adaptation with Deep Learning. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 2 (2018), 26.
[48] Yiting Xia, Xiaoye Steven Sun, Simbarashe Dzinamarira, Dingming Wu, Xin Sunny Huang, and TS Ng. 2017. A Tale of Two Topologies: Exploring Convertible Data Center Network Architectures with Flat-tree. In *SIGCOMM*.
[49] Zhenjie Yang, Yong Cui, Xin Wang, Yadong Liu, Minming Li, and Zhixing Zhang. 2019. Towards Maximal Service Profit in Geo-Distributed Clouds. In *IEEE ICDCS*.
[50] Jin Y Yen. 1971. Finding the K Shortest Loopless Paths in a Network. *Management Science* 17, 11 (1971), 712–716.
[51] Junlan Zhou, Malveeka Tewari, Min Zhu, Abdul Kabbani, Leon Poutievski, Arjun Singh, and Amin Vahdat. 2014. WCMP: Weighted Cost Multipathing for Improved Fairness in Data Centers. In *EuroSys*.
[52] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y Zhao, and Haitao Zheng. 2012. Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *SIGCOMM*.
[53] Yibo Zhu, Xia Zhou, Zengbin Zhang, Lin Zhou, Amin Vahdat, Ben Y Zhao, and Haitao Zheng. 2014. Cutting the Cord: a Robust Wireless Facilities Network for Data Centers. In *MOBICOM*.

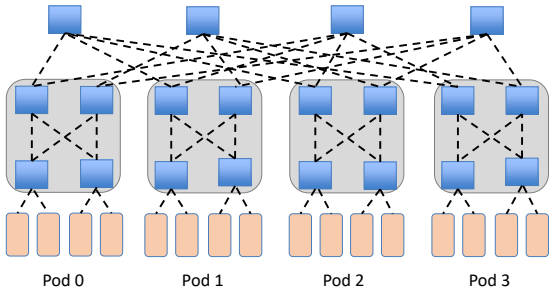


Fig. 9. Fat-tree topology with 4 pods

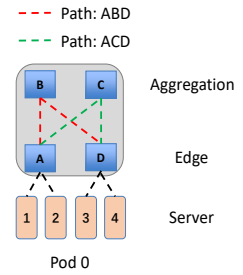


Fig. 10. Routing paths from **1** to **3**

A APPENDIX

A.1 Proof of Theorem 1

Our reduction is from the integer partition problem, whose task is to decide whether a given set $A = \{a_1, a_2, ..., a_n\}$ of positive integers can be partitioned into two subsets S_1 and S_2 such that the sum of numbers in S_1 equals the sum of numbers in S_2. For any integer partition problem instance with set A, we construct an instance of $\mathcal{P}_0$ as follows: first, we consider a 4-pod fat-tree network with all γ_{ij} equal to 0, as shown in Fig. 9. Each link on the network has a uniform link capacity, which is equal to or larger than the maximum number in set A. Second, we consider two servers, labeled by **1** and **3** in Fig. 10. They are in the same pod but under different edge switches. We set the former as the source of flows and the latter as the destination, thus flows can only choose from two paths (e.g., ACD and ABD) to reach the destination, as shown in Fig. 10. Suppose that there are n flows from the source node to the destination node and their demands are represented by each number in A. This instance of $\mathcal{P}_0$ can be constructed in polynomial time.

Assuming that there exists a successful partition in set A, i.e.,

$$\sum_{a_i \in S_1} a_i = \sum_{a_i \in S_2} a_i,$$

if we assign each subset of flow demands to one of the paths, it ensures the minimization of the maximum link utilization. If there is no successful partition in set A, unequal subsets would lead one of the paths to have under-loaded links while the other path faces congestion.

In the other direction, assuming that the network is perfectly load-balanced, i.e., the flows passing through the path ABD have the same total demand as the flows passing through the path ACD, a successful partition in set A is found at the same time. As the integer partition problem is well-known to be NP-hard, $\mathcal{P}_0$ is NP-hard, too. This completes the proof.

A.2 Proof of Theorem 2

We construct the approximation algorithm for $\mathcal{P}_0$ as follows. First, we use the ρ_1-approximation algorithm to solve $\mathcal{P}_1$. Denote the output IS solution as $\{e^*_{ij}\}$ and the output objective value as $\widetilde{\lambda}_1$. Let λ_1 denote the optimal objective value of $\mathcal{P}_1$, we have

$$\widetilde{\lambda}_1 \leq \rho_1 \lambda_1$$

Second, we set the available wireless link set $\overline{E}_s$ in $\mathcal{P}_2$ as $\{e^*_{ij}\}$. By relaxing $x^k_{ij} \in \{0, 1\}$ to $[0, 1]$, we obtain the relaxed problem of $\mathcal{P}_2$ and denote it as $\widetilde{\mathcal{P}}_2$. Let $\widetilde{\lambda}_2$ denote the optimal solution of $\widetilde{\mathcal{P}}_2$. Since any feasible solution of $\mathcal{P}_1$ with the same IS is a feasible solution of $\widetilde{\mathcal{P}}_2$

and vice versa, we have

$$\widetilde{\lambda}_2 \leq \widetilde{\lambda}_1$$

We use a ρ_2-relaxed algorithm to solve $\mathcal{P}_2$ and denote the output objective value as λ_2^*. As we call an approximation algorithm for an integer programming (IP) minimization problem ρ_2-relaxed if it can achieve a solution within ρ_2 times the optimal solution of its LP relaxation, we have

$$\lambda_2^* \leq \rho_2 \widetilde{\lambda}_2$$

Finally, since $\mathcal{P}_1$ is the LP relaxation of $\mathcal{P}_0$, let λ_0 denote the optimal solution of $\mathcal{P}_0$, we have

$$\lambda_1 \leq \lambda_0$$

Therefore, we have

$$\lambda_2^* \leq \rho_2 \widetilde{\lambda}_2 \leq \rho_2 \widetilde{\lambda}_1 \leq \rho_1 \rho_2 \lambda_1 \leq \rho_1 \rho_2 \lambda_0$$

This completes the proof.

A.3 Proof of Lemma 1

Consider an arbitrary link $e_{ij} \in G$. Let I denote a set of neighbors of e_{ij} that are *independent* from each other. Since there is at most one *independent link* at each rack position (consider the single-radio single-channel condition), the maximum number of *independent neighbors* generated from these rack positions is at most $J(\mathcal{F}(2\pi, r), \mathcal{G})$. It is easy to verify that, when θ is very small relative to D_0, we can turn all the transmission directions of these neighbor links to point at the node v_j thus conflicting with link e_{ij} while keeping *independent* with each other. Hence

$$|I| \leq J(\mathcal{F}(2\pi, r), \mathcal{G})$$

According to our definitions of $\mathcal{F}(\theta, r)$ and $J(\mathcal{F}(\theta, r), \mathcal{G})$, where $\mathcal{F}(\theta, r)$ is defined as the fan-shape interference range of the 60GHz link (a sector with an angle θ and a radius r) and $J(\mathcal{F}(\theta, r), \mathcal{G})$ is defined as the maximum number of racks that are located in the range of any sector $\mathcal{F}(\theta, r)$ with its center at a rack in $\mathcal{G}$, it can be easily proved that

$$J(\mathcal{F}(2\pi, r), \mathcal{G}) \leq \frac{2\pi}{\theta} J(\mathcal{F}(\theta, r), \mathcal{G})$$

This completes the proof.

A.4 Proof of Theorem 3

In $G_c(V_c, E_c)$, each node $v_i \in V_c$ has a weight $w(v_i)$. Let $w(V)$ denote the sum $\sum_{v_i \in V} w(v_i)$. Let V_i denote the set $\{v_1, v_2, ..., v_{i-1}\}$ following an ordering σ. We define V_i^+ and V_i^- as

$$V_i^+ = V_i \cap N_\ell^+(v_i)$$

and

$$V_i^- = V_i \cap N_\ell^-(v_i)$$

Based on [45], there exists a surplus node ordering σ^* satisfying $w(V_i^+) \geq w(V_i^-)$ for $1 \leq i \leq n$. Hence we have

$$\begin{aligned} & w(v_i) + w(\Gamma_{\sigma^*}(v_i)) \\ = \; & w(v_i) + w(V_i^+) + w(V_i^-) \\ \leq \; & w(v_i) + 2w(V_i^+) \\ \leq \; & w(v_i) + 2w(N_\ell^+(v_i)) \end{aligned}$$

For any vector $\mathbf{w} \in Q'$:

$$w(v_i) + w(\Gamma_{\sigma^*}(v_i)) \leq w(v_i) + 2w(N_\ell^+(v_i)) \leq 1$$

thus we have $\mathbf{w} \in Q$ and $Q' \subseteq Q$. Since $Q \subseteq P$, we have $Q' \subseteq P$. Consider an arbitrary $\mathbf{w}_I \in P$, i.e., $\mathbf{w}_I$ is the incidence vector of an independence set I in P. For any node $v_i \in V_c$, the v_i and its in-neighbours $N_\ell^+(v_i)$ can not appear in the same independence set I. Hence for the case that I contains v_i, we have

$$w_I(v_i) + 2w_I(N_\ell^+(v_i)) = 1$$

otherwise

$$w_I(v_i) + 2w_I(N_\ell^+(v_i)) = 2|I \cap N_\ell^+(v_i)| \leq 2\rho^+$$

where ρ^+ is defined as the maximum size of any independent set in $G_c[N_\ell^+(v_i)]$. To summarize, given any $\mathbf{w}_I \in P$, we have

$$w_I(v_i) + 2w_I(N_\ell^+(v_i)) \leq \max\{1, 2\rho^+\}$$

i.e., $\mathbf{w}_I \in \mu Q'$ where $\mu = \max\{1, 2\rho^+\}$. Hence $P \subseteq \mu Q'$.

According to the definition of our edge direction ℓ, if any link e_{uv} is an in-neighbor of link e_{ij}, then v_v must be within the interference range of v_i. Since the number of racks that are located within the interference range of v_i is at most $J(\mathcal{F}(\theta, r), \mathcal{G})$ and each rack position has at most one independent link (consider the single-radio single-channel condition), the number of independent in-neighbors is at most $J(\mathcal{F}(\theta, r), \mathcal{G})$. Based on the definition of ρ^+, we have

$$\rho^+ \leq J(\mathcal{F}(\theta, r), \mathcal{G})$$

This completes the proof.

A.5 Proof of Lemma 2

We first show that the expectation of total flows on any edge is less than or equal to the μ-approximation solution of $\mathcal{P}_1$. Let $\mathcal{I}$ denote the set of all the decomposed ISs and an empty set $I_\emptyset$. Let $X_I = 1$ denote the IS I in $\mathcal{I}$ is selected and otherwise $X_I = 0$. Hence we have $Pr(X_I = 1) = y(I)$ and $\sum_{I \in \mathcal{I}} y(I) = 1$.

For simplicity, we take x_e^k also as the flow demand $(x_{ij}^k + x_{ji}^k)$ of f_k on edge e. Let $z_e^k(I)$ denote the consumed flow demand of f_k on edge e when I is scheduled, we have

$$z_e^k(I) = \frac{d^k x_e^k y(I)}{\sum_{e \in I} y(I)}$$

Let X_e denote the total flow on edge e. Then we have

$$\begin{aligned} E[X_e] &= E[\sum_{e \in I} \sum_k z_e^k(I) X_I] \\ &= (\sum_k d^k x_e^k) \cdot (\frac{\sum_{e \in I} E[y(I) X_I]}{\sum_{e \in I} y(I)}) \\ &= \sum_k d^k x_e^k \cdot \frac{\sum_{e \in I} y^2(I)}{\sum_{e \in I} y(I)} \end{aligned}$$

Noticing that

$$\sum_{e \in I} y^2(I) \leq \sum_{e \in I} y(I) \leq \sum_{I \in \mathcal{I}} y(I) \leq 1$$

we have

$$E[X_e] \leq \sum_k d^k x_e^k$$

Suppose λ^* is the optimal objective of $\mathcal{P}_3$. Since the polytope Q' of $\widetilde{\mathcal{P}}_3$ is a μ-approximation of the polytope of $\mathcal{P}_3$, according to the constraint (10), we have

$$\sum_k d^k x_e^k \leq \mu\lambda^* \mathcal{C}_e \cdot \sum_{e \in I} y(I) \leq \mu\lambda^* \mathcal{C}_e$$

Hence

$$E[X_e] \leq \mu\lambda^* \mathcal{C}_e$$

holds. Let Y_e denote $\frac{X_e}{\mathcal{C}_e}$, we have

$$E[Y_e] \leq \mu\lambda^*$$

holds for all wireless edges $e \in \overline{E}_s$. As the link utilization of any wired edge $e \in E_w$ is always less than $\mu\lambda^*$ because of constraint (9), the expectation of link utilization on any edge $e \in E$ will not exceed the approximation solution $\mu\lambda^*$, i.e., μ-approximation solution of $\mathcal{P}_3$. Since $\mathcal{P}_3$ is the relaxation of $\mathcal{P}_1$, the lemma holds.

A.6 Proof of Theorem 4

Based on Lemma 1, we show that it is unlikely that the random variable Y_e deviates significantly from its expectation. Since we select exactly one element of $\mathcal{I}$, we have

$$\sum_{I \in \mathcal{I}} X_I = 1$$

Note that Y_e is the weighted sum of $\{X_I\}$ with a weight $w_I = \sum_k z_e^k(I)/\mathcal{C}_e$ for each X_I. To ensure the weighted coefficient $w_I \leq 1$, for any $w_{I^*} \geq 1$, we split such I^* as m equal ISs $\{I_j^*\}$ where each I_j^* is attached with a weight $w_I/m \leq 1$. Note that this does not change the expectation of Y_e and also ensures $Y_e = \sum_{e \in I} w_I X_I$ where $w_I \leq 1$ and $\sum_{e \in I} X_I \leq 1$. Hence based on [18], the variables $\{X_I\}$ satisfy the *negative-association* condition and the Chernoff bound holds for Y_e, i.e., we have

$$Pr(Y_e > \frac{6 \ln n}{\ln \ln n} \max\{1, \mu\lambda^*\}) < \frac{1}{n^3}$$

According to Boole's inequality, we have

$$Pr(\exists e \in E : \ Y_e > \frac{6 \ln n}{\ln \ln n} \max\{1, \mu\lambda^*\}) < |E| \cdot \frac{1}{n^3} < \frac{1}{n}$$

i.e., our algorithm achieves an $O(\frac{\log n}{\log \log n})$-approximation solution to λ^* with high probability. Finally, since $\mathcal{P}_3$ is the relaxation of $\mathcal{P}_1$, suppose λ_1 is the optimal solution for $\mathcal{P}_1$, we have $\lambda_1 \geq \lambda^*$. This completes the proof.

A.7 Proof of Theorem 5

According to Theorem 2 and Theorem 4, we can combine the two approximation ratios $O(\frac{\mu \log n}{\log \log n})$ and $O(\frac{\log n}{\log \log n})$ for $\mathcal{P}_1$ and $\mathcal{P}_2$ to construct the final approximation ratio $O(\mu(\frac{\log n}{\log \log n})^2)$ for $\mathcal{P}_0$. This completes the proof.

A.8 Proof of Claim 1

Constraints (5)(6)(7) guarantee that the routing path of f_k determined by the optimal solution of the reconfiguration problem is a single path from s_k to d_k, and constraints (16) guarantee that this optimal single path contains all links in the former path before the reconfiguration. Thus the existence of any extra link besides R_k will contradict the optimality, which completes the proof.

A.9 Proof of Theorem 6

Based on the randomized rounding theory in [18] and [38], and our definitions about ρ-approximation and ρ-relaxed algorithms in Section 2, lines 3-7 in FRS achieves a $O(\frac{\log n}{\log \log n})$-approximation solution for the problem $\widetilde{\mathcal{P}}_4$ with high probability. As $\widetilde{\mathcal{P}}_4$ is relaxed from $\mathcal{P}_4$, FRS can achieve an $O(\frac{\log n}{\log \log n})$-relaxed solution for $\mathcal{P}_4$ with high probability. According to our definitions for ρ-approximation and ρ-relaxed algorithms, FRS is also an $O(\frac{\log n}{\log \log n})$-approximation algorithm for $\mathcal{P}_4$ with high probability. This completes the proof.

Received July 2019; revised August 2019; accepted September 2019