Diamond: Nesting the Data Center Network With Wireless Rings in 3-D Space

Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Shenghui Yan, Chao Zhu, Xiang-Yang Li, *Fellow, IEEE*, and Ning Ge, *Member, IEEE*

***Abstract*— The introduction of wireless transmissions into the data center has shown to be promising in improving cost effectiveness of data center networks (DCNs). For high transmission flexibility and performance, a fundamental challenge is to increase the wireless availability and enable fully hybrid and seamless transmissions over both wired and wireless DCN components. Rather than limiting the number of wireless radios by the size of top-of-rack switches, we propose a novel DCN architecture, *Diamond*, which nests the wired DCN with radios equipped on all servers. To harvest the gain allowed by the rich reconfigurable wireless resources, we propose the low-cost deployment of scalable 3-D ring reflection spaces (RRSs) which are interconnected with streamlined wired herringbone to enable large number of concurrent wireless transmissions through high-performance multi-reflection of radio signals over metal. To increase the number of concurrent wireless transmissions within each RRS, we propose a precise reflection method to reduce the wireless interference. We build a 60-GHz-based testbed to demonstrate the function and transmission ability of our proposed architecture. We further perform extensive simulations to show the significant performance gain of diamond, in supporting up to five times higher server-to-server capacity, enabling network-wide load balancing, and ensuring high fault tolerance.**

***Index Terms*— Data center network, network architecture, millimeter-wave wireless communication.**

I. Introduction

THE high-performance data center network (DCN) is an essential infrastructure for cloud computing. There is a quick growth of large-scale services (e.g., Google Search, Hadoop, MapReduce, etc.) in the cloud, and recent measurements show tremendous traffic variations over space and time in DCNs [1]–[5]. In general, more than 80% flows in data centers are delay-sensitive small flows (e.g., queries or real-time small messages), while the majority of the traffic volume is contributed by the top 10% throughput-sensitive large flows (e.g., the backup traffic) [3], [4]. Recent work [6]–[9] introduces how to estimate the traffic patterns of data centers. In face of the uneven traffic distribution, conventional wired DCNs generally adopt the fixed and symmetric network design, which may lead to prevalent hot spots across different layers of the architecture and significantly reduce the performance of DCNs [1], [3], [10].

There are some recent interests on constructing *hybrid* DCNs [8], [9], [11]–[13] with the introduction of new network components such as optical circuit switches or wireless radios into the DCNs to provide configurable links [7], [14]–[20]. Although these hybrid infrastructures show the potential in achieving higher DCN capacity and lower transmission delay, their wired structures are kept unchanged even though they are not primitively designed to work with new network techniques, which limits the performance of hybrid DCN. Specifically, the new network components are added directly into conventional DCNs or are applied to replace part of existing network switches [8], [9], [12], [13]. Considering only the *local* performance improvement, it is hard for existing schemes to achieve the global optimal performance in the presence of network-wide traffic changes. The key challenge of a *fully hybrid* network design is to form a novel hybrid network architecture that can take full advantage of different network techniques and enable coherent and seamless transmissions for much higher DCN performance.

The low cost of today's commodity 60GHz radios makes their wide deployment a better option in a fully hybrid network design [9], [12], [13], [21]. Providing high wireless availability in the data center is the key to achieving high performance gain in a hybrid architecture. In existing proposals for hybrid DCNs, wireless radios are generally deployed on a flat 2D plane at the top of racks, which is susceptible to signal blocking [13]. Although a flat reflector on the room ceiling was proposed to alleviate the problem [11], [13], the ceiling height is quite restricted (3 meters [13]) and the method requires clearance above racks, which is usually infeasible in conventional data centers. The small rack size also restricts the number of radios that can be placed on each rack (at most eight radios per rack [12], [13]). If radios are densely deployed on top of racks, the strong interference between radios would restrict the number of concurrent wireless links, which in turn compromises the system performance [12]. The need of deploying more radios and links in the hybrid network for higher wireless availability calls for a completely new DCN architecture design.

Manuscript received May 28, 2016; revised September 15, 2017; accepted October 20, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Joo. Date of publication December 7, 2017; date of current version February 14, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61422206 and in part by the Tsinghua National Laboratory for Information Science and Technology. The work of X. Wang was supported in part by the NSF CNS under Grant 1526843 and in part by the NSF ECCS under Grant 1731238. The work of X.-Y. Li was supported in part by the China National Funds for Distinguished Young Scientists under Grant 61625205, in part by the Key Research Program of Frontier Sciences, CAS, under Grant QYZDY-SSW-JSC002, in part by the NSFC under Grant 61520106007, in part by the NSF ECCS under Grant 1247944, and in part by the NSF CNS under Grant 1526638. *(Corresponding author: Yong Cui.)*

Y. Cui, S. Xiao, Z. Yang, and S. Yan are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: cuiyong@tsinghua.edu.cn; xiaoshihan.xsh@gmail.com).

X. Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11790 USA (e-mail: x.wang@stonybrook.edu).

C. Zhu is with Aalto University, 02150 Espoo, Finland.

X.-Y. Li is with the University of Science and Technology of China, Hefei 230026, China.

N. Ge is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNET.2017.2773539

In this work, we propose a novel *fully-hybrid* network architecture, named Diamond, which ensures high wireless availability for efficient and high-performance DCN communications. Rather than restricting the radios to be on top of racks, we propose to deploy wireless radios along with a large number of servers. To avoid the interference among dense radios at the 2D plane, we propose to construct multiple **R**ing **R**eflection **S**paces (RRSs) to make the radios sparsely distributed in the 3D space. Inside each RRS, we develop a novel multi-reflection method to avoid the blocking of wireless links. As there is a limit on the transmission distance of 60GHz wireless signals, we propose a bucket-based modular topology design, where we limit the bucket dimension thus the ring range but increase the number of buckets when the size of the data center scales up. With our design, there is no need of changing the room plan above racks. Diamond has three key design features:

- **Novel hybrid network topology (§II)**: Rather than adding wireless radios directly on top of racks, we propose a fully hybrid network topology by constructing RRSs in Diamond to facilitate wireless transmissions and isolate the wireless interference. It also supports direct server-to-server wireless links rather than conventional rack-to-rack links. Then we apply a streamlined wired herringbone to interconnect the RRSs at low cost.
- **Precise multi-reflection of wireless links (§III)**: The susceptibility to blocking and the interference are two major issues that limit the wireless performance in DCNs. To the best of our knowledge, this is the first work that develops the multi-reflection transmission method to address the challenge of signal blocking. We further design a novel precise reflection scheme to efficiently restrict the wireless interference in the presence of a large number of concurrent wireless links.
- **Hybrid routing in scalable architecture (§IV & §V)**: The limited transmission distance of high-frequency wireless is the key challenge to the design of a large-scale data center. We propose a modular component to scale the Diamond architecture. In addition, we develop a greedy solution for the load balancing problem that proven to be NP-hard, and an opportunistic hybrid routing scheme to allow for low transmissions delay and graceful fault tolerance. We further show that the network diameter of Diamond can scale logarithmically with the server number to effectively bound the route length.

We implement a 60GHz-based testbed, and our experimental results confirm the high performance of multi-reflection, and demonstrate that proper reflection holes can efficiently reduce the interference in 3D space (§VII). Driven by the testbed parameters, our simulations show that Diamond can support up to five times higher server-to-server capacity and ensure graceful fault tolerance (§VIII). Finally, we introduce the related work (§IX) and draw the conclusions (§X).

II. Architecture

In this section, we first introduce the basic architecture and methodologies used in the Diamond system, and then present its hybrid topology design.

A. Diamond System Overview

At a high level, the Diamond system should meet the data center needs at different timescales. First, the configuration of wireless links should be updated periodically so that the network topology can better accommodate the current traffic of the data center. Second, given a configured network, we need to efficiently route the flows in real time.

Dynamic Wireless Configuration: Following the prior studies, the Diamond system exploits the controller of software-defined networking (SDN) for flexible and efficient configuration of the wireless links and routing paths [22]–[25]. More specifically, the Diamond controller periodically updates the configuration of the wireless links based on the traffic conditions reported from SDN-controllable ToR switches. Servers are equipped with high-capacity wireless radios (60GHz radio [9] or FSO transceivers [11]). To dynamically configure the wireless links, they are allowed to communicate with each other either directly by steering and aligning the antennas (physically or electronically driven [9], [11]) or using a multi-reflection method we propose. The controller first builds wireless links to alleviate the heavy traffic from the hot spots, and then randomly forms additional wireless links using the remaining available radios to achieve the benefits of random networking [26].

Hybrid Routing: The controller only computes the routing paths of hot-spot server pairs during the wireless configuration to alleviate the hot-spot traffic globally for the network-wide load balancing thus higher network throughput. For other light-loaded server pairs, the routing decision is made distributedly by servers and switches so that their traffic can go through available wireless links opportunistically to cut short the routing paths in real time.

B. Key Methodologies

There are two main challenges to implement a fully hybrid network: (1) When a large number of wireless links are enabled, the interference will restrict the number of concurrent transmissions; and (2) When a large number of wireless radios are deployed, the high-frequency wireless links are easily blocked by obstacles such as the supply pipes of air conditioning or the steel structures above racks. In light of these problems, Diamond applies a 3D deployment of the wireless radios to facilitate high number of concurrent wireless transmissions taking advantage of the following key techniques:

Space Division Multiplexing: To disperse the wireless radios, the radios in Diamond are installed with servers at different heights. Rather than deploying the wireless radios densely on only one flat 2D plane, we place the wireless radios on several separated large annular surfaces. Thus the deployment density of wireless radios is much lower than that of previous studies [12], [13]. The adjacent annular surfaces form a RRS where the signal can run from one radio to another. Due to the space division, the same set of wireless channels can be multiplexed across different RRSs, which helps isolate the interference in Diamond.

Multi-Reflection Transmission: Although more radios can be deployed in a 3D space, many radios cannot reach each other with existing direct point-to-point transmission or the one-reflection transmission [9], [11]–[13] due to the obstacle blocking. Instead, in Diamond, we utilize multiple reflections to bounce the signal emitted from one server to another. This helps to greatly increase the number of available wireless links. Following the prior work [13], our testbed experiment confirms that using the flat metal board as reflector can offer very good

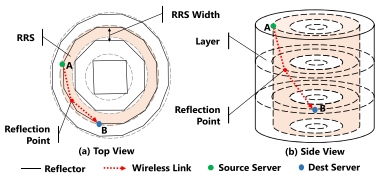


Fig. 1. Brief view of the wireless ring in Diamond (N = 4 rings and H = 4 layers).

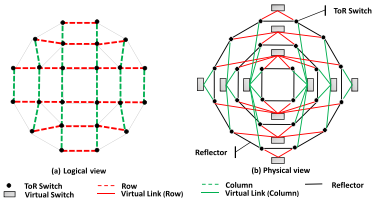


Fig. 2. Top view of wired herringbone in Diamond (N = 3 rings).

specular reflection with little energy loss or changing path loss during each reflection. This avoids the overhead of buffering and switching packets in multiple hops over intermediate switches.

Different types of directional antennas may have different beam widths [9]. For a multi-reflection path, there is a trade-off between the antenna beam width and the tolerance for the antenna alignment error. The narrower the beam width, the higher the antenna gains, but the less the alignment error tolerance. In the extreme case of using FSO with nearly zero beam width, previous study shows that using electrically-driven Galvo mirrors is possible to implement precise steering control [11]. For conventional 60GHz antennas, the electrically-driven antenna array is promising to satisfy this requirement [9], [12].

Precise Reflection Technology: Since the wireless antenna may have a wide beam width [9], multiple reflections would introduce unexpected interference inside the 3D space due to the signal leakage of the beam (e.g., the undesired side lobes of the 60GHz wireless beam [9], [21]). In order to efficiently restrict and control the interference caused by reflections, we develop a precise reflection method with the careful placement of absorbing materials on the reflection boards. Most areas of the board are covered by absorbing paper while small holes are left so that only the intended signal reflections are made by hitting the hole, which leads to very little signal leakage.

The basic *motivation* of the Diamond topology design is to enable more concurrent wireless transmissions. In our design, we separate the specific transmission functions of wireless and wired links in the network, so that both their distinct advantages on the transmission can be fully explored. We construct a ring-shape basic structure that enables wireless-only transmissions inside the ring employing the multi-reflections (§2.2). Then we apply the stable wired links to address the transmissions across different ring structures.

C. Topology Design

From the top view in Fig. 1, Diamond's topology is constructed by several concentric regular polygons with increasing radius. Polygons are numbered from inside to outside and named by *rings*, i.e., $\{R_i\}$, $1 \leq i \leq N$, where N is the total number of polygons. The i_{th} ring has $4i$ edges. The racks are placed at the vertex points of each ring, and there are totally $\sum_{1 \leq i \leq N} (4i) = 2(N^2 + N)$ racks, while flat metal reflectors are put at the edge of each ring. Rather than mounting the reflectors [11], [13] on the ceiling, reflectors in Diamond stand in perpendicular to the ground and have the same height as that of racks, which avoids the need of using clear ceiling space for wireless transmissions in data centers. In the following, we introduce the designs of major Diamond components.

Server and Rack: Each rack holds multiple servers at different height. The servers inside different racks at the same height form a *layer*, and the layers are numbered from the top to the bottom as $\{l_j\}$, $1 \leq j \leq H$. The height of each layer equals the height of a server at conventional racks, and the number of layers H equals to the number of servers in one rack. Therefore, a Diamond topology can accommodate totally $2(N^2 + N)H$ servers. Each server is equipped with 1 Ethernet port and 2 wireless ports with directional antennas. The networking principles in Diamond are: (1) the links between two servers are wireless; (2) the links between a server and its ToR switch or between two ToR switches are wired.

Wireless Links: The 3D space between two neighboring rings is called an RRS. For each server, one of its antennas points to RRS at its inner side and the other points to RRS at its outer side. By adjusting the antenna directions in the RRS, each server at ring R_i can flexibly communicate with other servers at different heights on rings R_i, R_{i-1} and R_{i+1} through direct transmissions or multiple reflections on different reflectors (Fig. 1 and Fig. 4).

Wired Links: With wireless links formed locally inside each RRS, the wired links are applied to interconnect different RRSs. Fig. 2 gives a top view of the wired connections in Diamond. Similar to conventional DCNs, the servers on each rack are connected to the common ToR switch. Fig. 2(a) shows the logical view of the *wired herringbone*. We number the horizontal lines in Fig. 2(a) from the top to the bottom as rows $\{r_i\}$, $1 \leq i \leq 2N$, and number the vertical lines from the left to the right as columns $\{c_i\}$, $1 \leq i \leq 2N$. Fig. 2(b) shows the physical connections of the wired herringbone. The principle of Diamond to interconnect the RRSs is that the ToR switches on the same row (or column) are interconnected by a *virtual switch*, while the ToR switches on different rows and different columns are not directly connected.

To implement the function of virtual switch, we have the option of applying any existing structure, e.g., the tree-based structure (Fat-tree [27]) or cube-based structure (BCube [28]), to interconnect the ToR switches on each row and each column. These structures may make the wired design of Diamond complex and costly. In Diamond, we prefer to apply the de-Bruijn graph [29] so that no additional switches are required. De-bruijn is attractive for providing constant link degree at each node and logarithmic network diameter. Then the path length is bounded and the routing structure is still simple (§IV). Although using de-Bruijn structure often involves complex wiring [30], the wiring is kept simple in Diamond because only one row (or column) of ToR switches are connected as one de-Bruijn.

D. Rack and Reflector Arrangement

There are two requirements to arrange the racks in Diamond to facilitate its practical and scalable deployment: first, all the reflector boards should be flat and have the same length to facilitate their economical production; second, the RRS width

should be kept stable, with the RRS width close to a fixed value when the number of rings increases. We call the physical distance between two neighboring rings R_i and R_{i+1} as the RRS width Δ_i (Fig. 1). Too large a RRS width will make Diamond occupy too much room area, while too small a RRS width will not leave enough space for wireless transmissions.

As mentioned earlier, all the polygons in Diamond are regular with the same edge length and are put concentrically in a symmetric way as shown in Fig. 2. The reflector height equals the height of racks, and the reflector length is denoted as L. Then our design ensures the RRS width Δ_i at ith ring to have the following property:

Property 1: $\lim_{i\to\infty} \Delta_i = 2L/\pi$

Proof: See the detailed proof in Appendix A of the online supplementary material. ■

Based on the above proof, the RRS width Δ_i decreases as the ring number i becomes larger. Property 1 ensures that the RRS width does not fall to zero but reaches a fixed limit value. For a setting L=2.5m, the RRS width Δ_i can keep a value close to the fixed limit value 1.6m. We can see that the RRS width becomes stable and approaches the fixed limit value quickly when the ring number increases, which demonstrates the scalability of the Diamond design.

III. Wireless Configuration

In this section, we first introduce our schemes of finding the reflection path when building a wireless link and eliminating the wireless interference during the reflections, and then present our strategies in forming flexible wireless configurations for network-wide load balancing.

A. Reflection Path

Since the physical topology of Diamond is fixed, the reflection paths can be easily calculated between any two servers. A reflection path table (termed the *cover table*) can be obtained offline at the initial deployment of Diamond. Given a source and destination server pair, if a reflection path can be found in the table, the antenna angles can be adjusted by the servers accordingly to build the wireless link. If there are multiple paths available between two servers, we choose the one with the least number of reflection times (direct transmission is considered as zero times of reflection).

To construct the cover table, we project the 3-dimensional coordinate of servers to a 2-dimensional coordinate by removing the height initially. For a given pair of servers, we first find the reflection path and the antenna angle in 2-dimensional coordinate with the Algorithm 1. We then transform the 2-dimensional angle to a 3-dimensional angle according to the height difference using the Algorithm 2.

The Algorithm 1 is applied to determine the set of reachable servers from a source server v_0 within n *reflection hops*. At the beginning (lines 1-2), we initialize the cover table with the reflector segments reachable in one hop by the server v_0 as illustrated in Fig. 3. The server v_0 can reach elements B_1 to B_5 directly in its first hop, and there are five *segments* of reflectors within its communication range. Let $\mathcal{S}_i$ denote the segment set that can be reached wirelessly at the ith hop from v_0, i.e., $\mathcal{S}_1 = \{s_{B1B2}, s_{B2B3}, s_{B3B4}, s_{B4B5}\}$.

To determine the next-hop segment set $\mathcal{S}_2$ based on $\mathcal{S}_1$, we first consider the first segment $s_{B1B2} \in \mathcal{S}_1$. Based on the law of reflection, we can easily obtain the segment s_{C1C2} reached by the reflection on s_{B1B2} (see Fig. 3),

Algorithm 1 CTC: **C**over **T**able **C**alculation

Input: Source server v_0, limited reflection hops n.
Output: Reachable server set R and corresponding path set P and antenna angle set Θ.
1: Set $\mathcal{S}_1$ as the reachable segments from v_0 at first hop.
2: $R \leftarrow \emptyset$, $v_0(s) \leftarrow v_0$ and $Parent(s) \leftarrow \emptyset$, $\forall s \in \mathcal{S}_1$.
3: **for** reflection hops i from 1 to n-1 **do**
4: **for** each segment $s \in \mathcal{S}_i$ **do**
5: Calculate the new reachable segment set $\mathcal{S}_{i+1}(s)$ by the reflection on segment s from $v_i(s)$.
6: Set $Parent(s^*) \leftarrow s$ for each segment $s^* \in \mathcal{S}_{i+1}(s)$.
7: Update $v_{i+1}(s)$ to be the point mirroring symmetry to $v_i(s)$ about segment s.
8: **end for**
9: $\mathcal{S}_{i+1} \leftarrow \cup_{s\in S_i}\{\mathcal{S}_{i+1}(s)\}$.
10: **end for**
11: **for** reflection hops i from 1 to n **do**
12: **for** each server $v_k \notin R$ covered in the $\mathcal{S}_i$ **do**
13: Calculate the initial antenna angle θ and path p from server v_0 to reach server v_k based on $\mathcal{S}_i$ and $Parent(\mathcal{S}_i)$.
14: $R \leftarrow R + v_k$, $P \leftarrow P + p$, $\Theta \leftarrow \Theta + \theta$.
15: **end for**
16: **end for**
17: **return** R, P, Θ.

Algorithm 2 3D-Space Signal Angle Calculation

Input: Initial 2D antenna angle θ^{2d}, the height difference H_{SD} between source server S and destination server D, and the path $p = (S, M_0, ..., M_{n-1}, D)$.
Output: The 3D antenna angle θ^{3d}.
1: Calculate H_0 with $\frac{H_0}{SM_0} = \frac{H_1}{M_0M_1} = ... = \frac{H_{n-1}}{M_{n-2}M_{n-1}} = \frac{H_n}{M_{n-1}D}$ and $H_0 + H_1 + ... + H_n = H_{SD}$.
2: Get inclination angle $\theta_z = arctan\ (\frac{SM_0}{H_0})$.
3: **return** $\theta^{3d} = \{\theta^{2d}, \theta_z\}$.

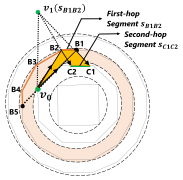


Fig. 3. Example of cover table calculation.

i.e., we have $\mathcal{S}_2(s_{B1B2}) = \{s_{C1C2}\}$ (line 5). For a segment s, we denote $Parent(s)$ as its previous-hop segment, i.e., $Parent(s_{C1C2}) = s_{B1B2}$ (line 6). Similarly, we can calculate the segment set $\mathcal{S}_2(s)$ for each $s \in \mathcal{S}_1$, and merge them as one set $\mathcal{S}_2$ (line 9). We can iteratively obtain the segment set $\mathcal{S}_{i+1}$ based on $\mathcal{S}_i$ until i equals $n-1$, and the segment sequences are cached in $Parent(\mathcal{S}_i)$ (line 3-10). Finally, for each server v^* covered by $\mathcal{S}_i$, we calculate the reflection path and the signal angle from the source server v_0 to reach v^* based on the segment set $\mathcal{S}_i$ and the segment sequence $Parent(\mathcal{S}_i)$ (line 11-13), and add it to the final cover table R as one reachable destination server from v_0 (line 14). As the antenna angle obtained by Algorithm 1 does not take

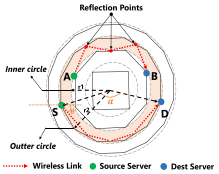


Fig. 4. Reflection paths in wireless rings.

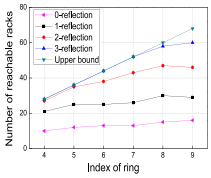


Fig. 5. Number of reachable racks per server at different rings and within different reflection times.

into account the height coordinate of server, in Algorithm 2, we further apply the equal-ratio reflection properties (line 1-2) to calculate the final third angle coordinate θ_z around the vertical axis based on the height difference.

In the following, we analyze the coverage of a reflection path with a limited number of reflections. To simplify the analysis, we consider a circle reflector instead for ring construction, termed the *circle case*. This is a reasonable approximation with similar coverage results because the original *polygon case* achieves a good approximation of a circle when the ring number is larger than 3 (see Fig. 1). In Fig. 4, the source server is assumed to be at the outer circle of a ring (e.g., the point S) and the maximum reflection times is n. The radiuses of the inner circle and outer circle of the ring are r_1 and r_2 respectively. Relative to the center of Diamond, the difference of the central angle between the source server and destination server is α. To reach the destination server, the antenna angle to transmit signal is θ. Then we have the following analytical results for the coverage properties for a reflection path within n reflection times:

Property 2: (a) Denote the maximum coverage at the inner circle as a central angle α_I, then

$$\alpha_I = \begin{cases} (n+1)\arccos\dfrac{r_1}{r_2}, & n = 2k, (k = 1, 2, 3\ldots), \\ n\arccos\dfrac{r_1}{r_2}, & n = 2k-1, (k = 1, 2, 3\ldots). \end{cases}$$

(b) Denote the maximum coverage at the outer circle as a central angle α_O, then $\alpha_O = 2(n+1)\cos\frac{r_1}{r_2}$.

(c) If the destination server is at the inner circle and $\alpha \leq \alpha_I$, the destination server is reachable and the antenna angle is

$$\theta = \begin{cases} \arcsin\dfrac{r_1 \sin\frac{\alpha}{i+1}}{\sqrt{r_1^2 + r_2^2 - 2\ r_1\ r_2 \cos\frac{\alpha}{i+1}}}, & i = 2k \text{ and } i \leq n, \\ \arcsin\dfrac{r_1 \sin\frac{\alpha}{i}}{\sqrt{r_1^2 + r_2^2 - 2\ r_1\ r_2 \cos\frac{\alpha}{i}}}, & i = 2k-1 \text{ and } i \leq n. \end{cases}$$

where i is the reflection times.

(d) If the destination server is at the outer circle and $\alpha \leq \alpha_O$, the destination server is reachable and $\theta = \arcsin\dfrac{r_1 \sin\frac{\alpha}{2(i+1)}}{\sqrt{r_1^2 + r_2^2 - 2\ r_1\ r_2 \cos\frac{\alpha}{2(i+1)}}}$, where i is the reflection times.

We omit the proofs since they are all geometry properties that can be directly derived from the static topology. To illustrate the specific coverage results in the Diamond topology, we simulate the reflection paths between all the server pairs based on Algorithm 1 and 2. Fig. 5 shows the average communication range of a wireless radio, i.e., the reachable rack number at both its current ring R_i and inner ring R_{i-1}. We can see that no more than three reflections can cover above 90% racks in the RRS when the ring number is less than 9. For a ring number larger than 4, a server can reach at least 10 racks through the direct transmission, 20 racks within a single reflection and 28 racks within two reflections. If we limit the reflection times of a path to be less than two, for a medium-size data center with 10000 servers, there will be more than 1 million potential wireless links available for use. The rich wireless links contribute a lot to the network-wide adaptive topology formulation and can support efficient routing and fault-tolerance in Diamond.

B. Reduction of Wireless Interference

We design a precise reflection method to alleviate the wireless interference during reflections. Specifically, we carefully place the absorbing materials on the reflection board and leave small *holes* for only the intended reflection points. In the following, we analyze the density and distribution of reflection points (i.e., the reflection holes) on the reflector boards.

To simplify the analysis, we first present a special *circle case* where the flat reflectors are replaced by curved reflectors so that all the polygons are transformed to their circumcircles. We consider the communication of servers inside the kth RRS, i.e., the communication between a server on ring k and another on ring $k+1$ and the communication between two servers on ring $k+1$. The reflection times are limited within three. The communication of servers in different rings is achieved by zero and double reflections. The double reflection forms the reflection points on the outer side of ring k and the inner side of ring $k+1$.

Considering the distribution of reflection points on ring k, we have the following property:

Property 3: At each layer of Diamond, for an arbitrary reflector on ring k, there are at most six reflection points on the reflector board.

Proof: See the detailed proof in Appendix B of the online supplementary material. ■

We obtain the expressions of the central angle for each reflection point in ring $k+1$ following the same procedure of ring k. We examine the distribution of reflection points on each reflector in ring $k+1$ based on simulation results, and found that at each layer from the ring 5 to the ring 50, there are average ten reflection points on the board of the ring $k+1$. One hole may be reused by a large number of reflection points for different reflection paths, i.e., the distance between two reflection points is small enough to overlap with each other. With the reuse ratio equal to the ratio of reused points to the total number of reflection points, Fig. 6 shows that the reuse ratio is high and increases when the ring number becomes larger.

C. Configuration for Hot-Spot Traffic

Since the above techniques enable a large number of server-to-server wireless links, Diamond can implement a network-

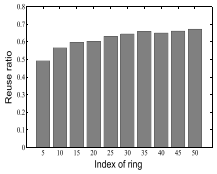


Fig. 6. Reuse ratio of reflection points on a board at different rings.

wide reconfigurable topology for balancing the identified hot-spot traffic, which contributes to high throughput and effective routing.

1) Configuration Problem: We denote the topology of a *hybrid* DCN as a graph $G(V, E)$, where V denotes all the nodes including servers and switches, and E denotes all the links connecting the nodes. Specially, we denote the wired link set in E as E_w, and denote E_s as all the possible wireless links in E that can be created. In Diamond, as the wired links only exist between the server and switch while the wireless links only exist between servers, we have $E_s \bigcap E_w = \phi$ and $E = E_s + E_w$. At any moment, the network topology of Diamond is a subgraph of $G(V, E)$ where its wired link set is E_w, and its wireless link set is a subset contained in E_s.

Given a hybrid topology $G(V, E)$, we construct its corresponding *conflict graph* $G_I(V_I, E_I)$ to describe the interference relations among all the wireless links E_s. We consider each wireless link $e \in E_s$ as a vertex in G_I, and add an edge (e_1, e_2) between any two vertices e_1 and e_2 if and only if they conflict with each other. The conflict graph can be obtained offline by applying the measurement method in [9]. An *independent set* (IS) in a conflict graph $G_I(V_I, E_I)$ is defined as a vertex subset of V_I where any two vertices are not directly connected. Hence a feasible solution for setting up wireless links in the topology G is to select an IS in the corresponding conflict graph G_I.

The computation of wireless configuration is performed by the network controller in DCNs. The controller input is a traffic demand matrix with each entry being the traffic demand among a pair of servers. Suppose the input matrix includes $|K|$ pairs of servers and each server pair $k \in K$ is attached with a traffic demand D^k. The link capacity of link e_{ij} is denoted by $\mathcal{C}_{ij}$. Let δ_{ij}^p denote whether a path p contains link e_{ij}, and $P(k)$ denote the path set that contains all the possible paths of server pair k. When given the network topology $G(V, E)$, δ_{ij}^p and $P(k)$ are all initialized as specific values with respect to all the possible paths in G. Moreover, we use a binary variable f_p^k to denote whether path p is assigned to route the requested flow demand D^k, and a binary variable γ_{ij} to denote whether we will build the wireless link e_{ij}.

The objective of our network design is to minimize the maximum link utilization λ of the entire network during each scheduling period T. The link utilization of link $e_{ij} \in E$ is $\frac{\sum_{k\in K}\sum_{p\in P(k)} D^k f_p^k \delta_{ij}^p}{\mathcal{C}_{ij}T}$. Eq. (1) is the link capacity constraint, where the factor γ_{ij} on the right hand ensures that, for a feasible routing solution, a flow can use a wireless link only if it has been built. Eq. (2) and Eq. (3) are the flow constraints that ensure only one path is assigned to each flow. Eq. (4)(5) are appended to take into account the wireless interference. By solving HLB, the controller outputs an IS for the setup of wireless links.

Algorithm 3 HDF: **H**ighest **D**emand **F**irst

Input: Flow set F.
Output: Wireless link set $\mathcal{W}$ and routing path set $\mathcal{P}$.
1: **for** each flow $f_k \in F$ **do**
2: **for** each link $e \in P(k)$ **do**
3: Set link weight $w(e, f_k) \leftarrow D^k/len(e)$.
4: **end for**
5: **end for**
6: $w'(e) \leftarrow \max_{e\in P(k)} w(e, f_k)$ and $\mathcal{W} \leftarrow \{e \in E_s : w'(e) > 0\}$.
7: **while** $\mathcal{W} \neq \emptyset$ **do**
8: Select the link $e^* \in \mathcal{W}$ with highest weight $w'(e^*)$ to build.
9: Remove the links that conflict with e^* in $\mathcal{W}$.
10: **end while**
11: **for** each flow $f_k \in F$ in the descending order of D^k **do**
12: Calculate the path capacity C_p for its shortest-path set $\mathcal{P}^k$.
13: Distribute the traffic to path set $\mathcal{P}^k$ in proportional to C_p.
14: Update the remaining path capacity C_p for $\mathcal{P}^k$.
15: **end for**
16: $\mathcal{W} \leftarrow \{e \in E_s : e \in \mathcal{P}^k, f_k \in F\}$.
17: **return** $\mathcal{W}$, $\mathcal{P}$.

$$Min \ \ \lambda, \ \ \text{subject to}$$

$$\begin{cases} \sum_{k\in K}\sum_{p\in P(k)} D^k f_p^k \delta_{ij}^p \le \lambda \gamma_{ij} \mathcal{C}_{ij} T, \ \ \forall e_{ij} \in E & (1) \\ \sum_{p\in P(k)} f_p^k = 1, \ \ \forall k \in K & (2) \\ f_p^k = 0 \text{ or } 1, \ \forall k \in K, p \in P(k) & (3) \\ \gamma_{ij} + \gamma_{uv} \le 1, \ \forall \ (e_{ij}, \ e_{uv}) \in E_I & (4) \\ \gamma_{ij} = 0 \text{ or } 1, \ \forall e_{ij} \in E_s & (5) \end{cases}$$

The first hardness of HLB problem comes from the flexibility of load balance and routing in the hybrid topology. Here we generate the following theorem for the HLB problem:

Theorem 1: The formulated HLB problem is NP-hard.

Proof: See the detailed proof in Appendix C of the online supplementary material. ■

The second hardness of HLB problem comes from the complex wireless interference caused by the high flexibility of forming directional wireless links 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 [31]. Moreover, the small interference angle would lead to exponentially many ISs and consequently high computation complexity to look for the optimal solution. There are some existing studies [32]–[37] on finding an approximate solution for IS in some special interference graphs. In the following, in order to make Diamond support various types of antennas (e.g., 60GHz radio, FSO transceivers, etc.), we turn to the development of a fast heuristic solution for a general interference graph.

2) Greedy Scheduling: We design a greedy algorithm HDF (Highest Demand First, Algorithm 3) to provide a faster and simpler solution for HLB. The algorithm assigns a weight value to wireless links related to the flows (line 1-6), and then selects a set of non-conflict wireless links that maximize the

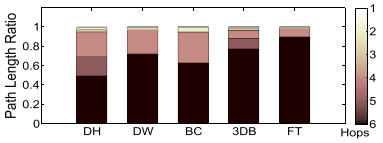


Fig. 7. Path length ratio of different topologies.

sum of weights (line 7-10). We define the weight of a wireless link as the ratio between the flow demand and the link length. For a link with reflections, the link length is the total geometric length of the reflection path. The intuition is that a link can provide larger benefit when serving higher flow demand over a shorter link length, as a shorter wireless link allows for smaller interference range and higher SNR thus higher link capacity. We greedily select the links with the largest weight to build first (line 8) and remove the links that conflict with the selected links (line 9). Next, the traffic demands are split into their shortest paths. Denote the minimum remaining capacity of links along a path as the *path capacity*. The server pair with the highest demand first splits the traffic to transmit over the set of shortest paths in proportional to the path capacity (line 12-13). Then the remaining link capacities are updated and the procedure repeats until no server pair is left (line 14). The gap between HDF and the optimal solution of HLB is evaluated in §VIII.

3) Random Networking for High Capacity and Low Delay: Since the wireless resources are rich in Diamond, after offloading the hot-spot traffic by HDF, some wireless radios may be left unused, particularly when the number of hot spots in the network is not big in a scheduling period. Random networking is shown to have the features of small average path length, high path diversity and high server-to-server network capacity [26], [38]. Thus we expect that the random formulation of wireless links helps to shorten the path length in Diamond. To verify this effect, we compare the percentage of the path length for all the server pairs under different DCN topologies with 512 servers in Fig. 7. DH is for Diamond where wireless links are built randomly; DW is for Diamond with the wired connections only; BC is the BCube topology [28]; FT is the Fat-tree topology [27]; 3DB is a Fat-tree topology augmented by 3D-beamforming radios at ToR switches [13]. We can see that the number of long paths in DW is larger than that in BCube. However, when introducing random wireless links, the ratio of short paths in DH is higher than all the other topologies. The short path length generally implies small hop delay and high end-to-end throughput due to fewer congestion points at intermediate routing hops [26], [38]. To benefit from the random networking in Diamond, we extend the IS selected by the HDF algorithm to a *maximal* IS, named the MIS, by randomly adding additional wireless links into the IS without creating conflict until no such kind of wireless link is available. The random formulation of wireless links in Diamond avoids the problem of complex wiring and costly management appearing in the previous work on using random wired links in DCNs [38].

IV. Routing Design

Diamond is built upon a topology-adaptive network, while existing routing protocols often impose a relatively long convergence time when the topology changes [39]. For more efficient routing, we propose to use a set of strategies in Diamond.

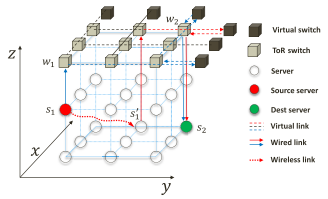


Fig. 8. Opportunistic hybrid routing in Diamond.

A. Overall Scheme

The setup of wireless links is performed by the Diamond controller periodically, with the length of period determined based on the traffic patterns learnt from historical records. For examples, the traffic load will change during different time of a day [14], so the setup can be adapted accordingly within a day and rerun daily. At the beginning of each period, a set of operations will be performed as follows: (1) The controller computes the wireless configuration and the routing paths for hot-spot server pairs using the methods described in §III, and sends out the instructions to both servers and their associated ToR switches. (2) The servers receiving the configuration instructions will adjust their antenna directions accordingly.

To summarize, there are three choices for a server or ToR switch to route its traffic. First, a server or ToR switch tries to match the routing rules designated by the controller. If matched, it delivers the packet accordingly. This first choice helps to balance the hot-spot traffic following the controller's decisions. Otherwise, it opportunistically utilizes its available wireless radios (if it is a server) or the available radios on its rack (if it is a ToR switch) to create a short-cut hop to the destination. This second choice contributes to shorter routing path through opportunistic hybrid routing (section §IV-C). If no wireless radios are proper to use, it delivers the packet to the next-hop node following a default wired routing path (section §IV-B). This last choice efficiently bounds the worst-case performance by routing through the wired herringbone.

B. Default Wired Routing

For the Diamond topology introduced in §II-C, a 3-tuple (x, y, z) labels a server at the xth row, yth column and zth layer. For simplicity, we use a 3-tuple $(x, y, 0)$ to label a ToR switch on the xth row and yth column. Fig. 8 shows a simple example to route from an arbitrary source server $s_1 = (x_1, y_1, z_1)$ to a destination server $s_2 = (x_2, y_2, z_2)$. Let $w_1 = (x_1, y_1, 0)$ and $w_2 = (x_2, y_2, 0)$ denote their corresponding ToR switches respectively. The shortest wired routing path can be established as follows. First, the packet routes from s_1 to w_1 and then we change one of the two coordinates of source ToR switch w_1 at a time to match that of switch w_2: $(x_1, y_1, 0) \rightarrow (x_2, y_1, 0) \rightarrow (x_2, y_2, 0)$. Finally, the packet routes from w_2 to s_2. Note that each coordinate change corresponds to hops through a virtual switch.

Suppose we apply de-Bruijn structure to implement the virtual switch, and the Diamond topology has totally $H = 2p$ layers and N rings. Then we need $4p$ ports per ToR switch, where $2p$ ports connect to the servers on the rack and $2p$ ports are used for constructing the de-Bruijn on its row and column. Since the diameter of a de-Bruijn graph is $\log_p N$, the path length through a virtual switch (i.e., the path length between two ToR switches on one row or column) can be bounded by

$\log_p N$. Based on the above routing procedure, we have the property:

Property 4: The network diameter, which is the longest shortest path among all the server pairs, of Diamond is bounded by $2\log_p N + 2$.

Since the Diamond with H layers and N rings can support totally $n = 2(N^2 + N)H$ servers, we have the diameter of Diamond as $O(\log_p n)$. Compared to conventional approaches (e.g., the Fat-tree [27] or VL2 [6] topology) which has a constant diameter but the number of switch ports increase with the number of servers, Diamond has much better scalability. As the server number increases, its network diameter extends logarithmically while the port number can be kept as a constant. This is similar to the recursion-based DCN topology such as BCube [28] and DCell [30]), which also has a logarithmic diameter when keeping a constant number of switch ports.

C. Opportunistic Hybrid Routing

The wired herringbone of Diamond provides the basic assurance of the connectivity and route length bound. Now we integrate the wireless transmissions into the default wired paths opportunistically. Suppose a server $s_1 = (x_1, y_1, z_1)$ receives a packet for a server $s_2 = (x_2, y_2, z_2)$, and the ToR switches of s_1 and s_2 are $w_1 = (x_1, y_1, 0)$ and $w_2 = (x_2, y_2, 0)$. The server s_1 is equipped with two radios, which are pointed to servers $s_1' = (x_3, y_3, z_3)$ and $s_1'' = (x_4, y_4, z_4)$, respectively. Define a *hamming distance* $\mathcal{D}(s_1, s_2)$ as the number of the unmatched coordinates between the tuples s_1 and s_2. Then the value range of $\mathcal{D}(s_1, s_2)$ is $\{0, 1, 2, 3\}$.

To perform the *opportunistic hybrid routing* (OHR), each server in Diamond simply follows two steps for the packet forwarding: (1) Call $d_1 = \mathcal{D}(s_1, s_2)$. If all three are matched (i.e., $d_1 = 0$), then it is the destination server. (2) Call $d_1' = \mathcal{D}(s_1', s_2)$ and $d_1'' = \mathcal{D}(s_1'', s_2)$. If $d_1' < d_1$ or $d_1'' < d_1$, forward the packet to the server s_1' or s_1'' accordingly through a wireless radio. Otherwise, forward the packet to the switch w_1 by default.

Similar to the servers, each ToR switch in Diamond forwards the packet as follows: (1) Call $d_1 = \mathcal{D}(w_1, s_2)$. If the first two are matched (i.e., $d_1 = 1$), it forwards the packet to s_2 directly; Otherwise, it randomly chooses one coordinate among the unmatched ones. Assume that ToR switch picks x_1 where $x_1 \neq x_2$, then the default next hop is $w_f = (x_2, y_1, 0)$. (2) For each server s_i in the rack, suppose its wireless radios point to two servers s_i' and s_i''. Call $d_i' = \mathcal{D}(s_i', s_2)$ and $d_i'' = \mathcal{D}(s_i'', s_2)$. If $d_1' < d_1$ or $d_1'' < d_1$, forward the packet to the sever s_i in the rack; Otherwise, forward it to the ToR switch w_f by default.

D. Fault-Tolerance

In Diamond, there exist redundant paths between any pair of servers, which makes it attractive for fault-tolerance. There are two types of failures to handle in Diamond: node failure and link failure. A node failure can be due to switch failure, server failure or wireless radio failure. A link failure will be resulted from a node failure or the impact of the environment. For example, wireless communications can be blocked due to the human movement in the RRSs. Clearly, due to the nested structure of Diamond, any single node or link failure does not lead to the network disconnection. We describe how a link failure is handled, as node failures can trigger the same responses.

In Diamond, each server has three different output links to forward the packets: (a) forward to the ToR switch it connects to; (b) forward to one of its two wireless radios. When a server finds one of its output links fails, it removes that wired/wireless connection from its connection list, and chooses one of the remaining available output links as its next hop based on the routing rules described in Section IV-C. In case the failure occurs at a wireless radio, the flows can be served by the alternative radio located in the same server to keep the flexible wireless link. If both radios in a server break down, the wired link can be selected in the current hop and the remaining routing hops will still follow the scheme of opportunistic hybrid routing in Section IV-C. The reflector failures may also affect flow transmissions. Taking advantage of the flexibility of wireless links, the network controller could adjust the antenna angles of wireless radios to configure the new reflection paths excluding the problematic reflectors. Benefited from the distributed routing property of OHR, the routing paths can be recovered quickly in Diamond to ensure high fault tolerance.

V. Scalability Design

So far, we can see the Diamond architecture can be scaled by adding more wireless rings around the same center with increasing diameters. However, a single Diamond with very large diameter is not easy to build and deploy, as compared to the conventional shipping-container-based, modular data center (MDC) design [28]. Moreover, when the network size further increases, the transmission performance of the outside rings will decrease. As the maximum wireless transmission distance is limited, each wireless link can only reach a small fraction of the servers in the ring when the ring diameter becomes very large. To further scale the architecture following the MDC design principle, we present a modular structure of Diamond, named the *Bucket*, that is convenient to build and deploy in large-scale data centers.

A. Bucket-Based Topology Design

Suppose the maximum wireless transmission distance is d_m. Based on the Diamond topology, we denote the *wireless efficiency* per ring as the ratio d_m/l_i, where $l_i = (\cot\frac{\pi}{4i})\pi L$ is the perimeter of the ith ring, and L is the length of the reflector board. Intuitively, when the ring perimeter increases at a higher index number, a single wireless link will cover a smaller fraction of the ring thus reducing the wireless efficiency. When the wireless efficiency is lower than a threshold η, the performance of in-ring wireless transmissions will be worse than using static wired links, which is against our purpose of exploiting wireless for better in-ring transmissions. To ensure the wireless efficiency inside the architecture, we propose to construct a basic modular structure with only a limited number of rings, termed the *bucket*. For the wireless efficiency to be higher than η, the number of rings in a bucket will be $\beta = \lfloor \frac{\pi}{4\arctan\frac{\pi\eta L}{d_m}} \rfloor$. Hence a bucket can hold at most $2(\beta^2+\beta)$ racks. To ensure enough wireless coverage in the presence of limited communication range of 60GHz signals, data center operators can restrict the number of rings within a bucket and build more buckets to scale up the wireless coverage in large data centers.

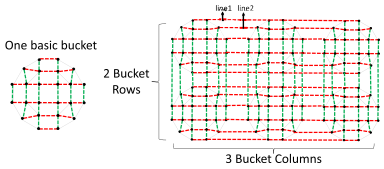


Fig. 9. A horizontal scaling example of Diamond: 2 rows and 3 columns of Buckets.

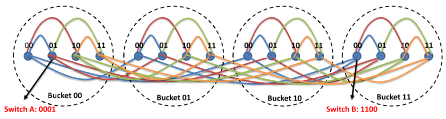


Fig. 10. Scalable routing example on one line (row or column).

When scaling the network with more racks, we construct more buckets and interconnect them using the wired links as Fig. 9 shows. Generally, the buckets are placed and connected logically as several rows and columns, while each bucket row (or column) contains multiple *lines* of racks (see Fig. 9). More specifically, the racks on the same line inside all the buckets are interconnected by a virtual switch, which can be considered as a horizontal extension of the previous Diamond structure. However, the previous use of de-Bruijn graph to implement the virtual switch is not suitable for a modular architecture, because a high-level de-Bruijn graph of buckets does not lead to a de-Bruijn sub-graph of racks inside each bucket. To address this modularity issue, we propose to implement the virtual switch by a new graph structure, named the *hamming graph* [40], where both inter-bucket and intra-bucket connections are following the same graph structure (see Fig. 10). As we will show in the next subsection, the use of hamming graph to establish connections on each line can ensure the network diameter to scale with the number of servers at a logarithmic factor, which provides a good scalability for the entire network architecture. With the modular bucket structure, Diamond will be more scalable and can be easily applied to construct a large-scale data center while ensuring a lower bound of the wireless efficiency.

B. Wired Routing Design

The routing strategy inside a single bucket has been introduced in section §IV.[1] Now we will introduce the routing between two servers located in different buckets. Suppose the entire network has N rows and M columns of buckets. Let B_{ij} denote the bucket at the ith row and jth column. A ToR switch located at the lth row and rth column of the bucket B_{ij} can be denoted as $Switch(i, j, l, r)$. In the following, we first illustrate the routing strategy using only wired links for communication between two ToR switches in different buckets, and then present the final hybrid routing strategy using both wireless and wired links for two servers in different buckets.

Suppose a data center has n rows and m columns of buckets. Inside each bucket, the maximum number of racks on one row or column is at most a. Given any integer x, let $b(x)$ be the binary string of x. We assign each rack $Rack(i, j, l, r)$ an ID as a connected binary string $b(i)b(j)b(l)b(r)$. Fig. 10 shows an example of the detailed wired connections as a hamming graph on one row, i.e., the second row in Fig. 9. For simplicity, we give each switch an ID $b(j)b(r)$. As Fig. 10 shows, the second switch A of the row inside the first bucket has an ID 0001, while the first switch B in the fourth bucket has an ID 1100. Based on the graph properties, we can see that each switch is only connected to the switches that have an ID differs from its own ID with only one bit. For example, the switch 0001 is connected directly with the switches 0000, 0011, 0101 and 1001. Therefore, we can change any bit of the switch ID to get the next-hop switch easily. For example, one shortest routing path between the switch A and switch B in Fig. 10 is $0001 \rightarrow 0000 \rightarrow 0100 \rightarrow 1100$. Now we turn to the general case in Fig. 9 where each switch has a full ID $w = b(i)b(j)b(l)b(r)$. We denote a function $\mathcal{F}(w_1, w_2)$ that takes the two switch IDs as its input and outputs the number of different bits in the two IDs. According to the above analysis, it is exactly the length of the shortest path between them. Then we have the following theorem:

Theorem 2: The network diameter, which is the longest shortest path among all the server pairs, of the *bucket-based* Diamond is bounded by $O(\log N)$, where N is the total number of servers in the network.

Proof: See the detailed proof in Appendix D of the online supplementary material. ■

C. Hybrid Routing Design

The above theorem provides the basic performance guarantee using only wired links. We will introduce how to integrate the flexible wireless links inside the buckets with the wired routing to further improve the performance. In Algorithm 4, we present the distributed routing strategy (SBR) for each server and ToR switch when receiving packets. We always try

Algorithm 4 SBR: **S**calable **B**ucket **R**outing

Input: Reachable server set R.

Output: Next-hop routing node v_x when a server or switch receives any packets.

Switch_Event_Handler

//triggered when receiving packets towards v_d

//suppose the current switch ID is $\widehat{w}$

1: Send the packets to v_d directly if $\widehat{w}$ is $w(v_d)$ and finish.
2: Randomly inverse one bit in $b(\widehat{w})$ that is different in the corresponding bit of $b(w(v_d))$ to get a new switch ID w'.
3: $v_x \leftarrow w'$, and send the packets to v_x.

Server_Event_Handler

//triggered when receiving packets towards v_d

4: $v_x \leftarrow \emptyset, D_{min} \leftarrow \infty$.
5: **if** the wireless radio is available **then**
6: **for** each next-hop node $\widehat{v}$ in the cover table R **do**
7: Route the packets to v_d directly if $\widehat{v}$ is v_d and finish.
8: **if** $\mathcal{F}(w(\widehat{v}), w(v_d)) + 2 \leq D_{min}$ **then**
9: $v_x \leftarrow \widehat{v},\ D_{min} \leftarrow \mathcal{F}(w(\widehat{v}), w(v_d)) + 2$.
10: **end if**
11: **end for**
12: **end if**
13: **if** $\mathcal{F}(w(\widehat{v}), w(v_d)) + 1 \leq D_{min}$ **then**
14: $v_x \leftarrow w(\widehat{v}),\ D_{min} \leftarrow \mathcal{F}(w(\widehat{v}), w(v_d)) + 1$.
15: **end if**
16: If there are multiple choices of v_x with same D_{min} in line 9 and line 14, randomly choose one as v_x to send packets.

[1]Since the opportunistic hybrid routing in §IV-C does not rely on specific implementations of the virtual switch, it can be applied with either the use of de-bruijn graph or hamming graph inside a single bucket.

to choose the next-hop that has a shortest wired path length to the destination server. Let $w(v)$ denote the ToR switch of the rack that holds the server v, and v_d denote the destination server. First, for each switch $\widehat{w}$, it randomly inverses one bit in its ID $b(\widehat{w})$ that is different to the corresponding bit in switch ID $b(w(v_d))$ to get the new ID as its next-hop switch (line 2). In this way, the number of different bits between $b(\widehat{w})$ and $b(w(v_d))$ is decreased by one at each hop, and the destination switch $w(v_d)$ is arrived when the bit difference is reduced to zero. Second, for each server v, it first checks all its reachable servers with wireless to find the one that has a shortest wired path to the destination server v_d, where the shortest path length is stored in D_{min} (line 5-12). Next, it checks the path length if choosing its ToR switch as the next-hop node (line 13-15). Among all the choices with the same shortest path length (including the wireless and wired next-hops), it randomly chooses one as the next-hop node v_x to achieve the load balance (line 16). As different data centers run different applications with different requirements on service performance guarantees, data center operators can assign links with different weights, i.e., giving wireless links higher preferences as compared to wired links or the other way. The shortest path algorithm in Algorithm 4 can be easily extended to provide the weighted shortest paths.

VI. Discussion on Deployment Issues

Circle vs. Polygon Reflector

We have so far suggested using the flat mental board as the reflector to facilitate its economic production and easy deployment. If the cost is not a concern, however, a curved mental reflector would allow the wireless communication range of each server to be larger than that of the flat reflector given the same constraint on reflection times. In order to determine the difference in communication coverage between using flat reflectors and using circular ones, we consider an example case where the ring number varies from 5 to 100 and the reflection times are set to be within three. We find that the average wireless communication range per server in the polygon case is above 80% that of the circle case. When the number of rings is smaller than 5, the servers of the entire ring can communicate in both cases. As the range gaps per server in these two cases are small and the cost of producing circular reflectors is much higher than that of flat ones, it is a better choice to select flat reflectors for the deployment in a large-size data center. In addition, Diamond is easier to deploy compared to some existing hybrid DCN proposals. The DCNs that use ceiling mirrors to build wireless connections require the clearance over racks, while it is expensive to remold the existing data centers. To reduce the reflector cost and the installation efforts, Diamond uses identical flat reflectors and places them on the ground. As the racks in current data centers are placed in a "column-row" style, they can be conveniently rearranged to form the columns and rows of Diamond shown in Fig. 2.

Design of Virtual Switch

Diamond introduces a virtual switch to interconnect the ToR switches on a line (row or column) and the virtual switch can be implemented by any existing interconnection structures, e.g., the tree-based structure [6], [27] or cube-based structure [28], [41], with different trade-offs between the cost and performance. However, the number of ports required by a virtual switch on different rows and columns may not be the same in Diamond. Consider a Diamond topology with $2n$ rows and $2n$ columns, the port numbers of virtual switch from row r_1 to r_n are $\{2, 4, 6, ..., n-2, n\}$. The uneven port numbers make it difficult to deploy conventional interconnection structures as some structures do not scale continuously [27], [28], [30]. To address this issue, we suggest using one virtual switch to interconnect two rows (or two columns) together to make a balance of the port number. Then each virtual switch requires $n+2$ ports by combining every two rows as (r_1, r_n), (r_2, r_{n-1}), (r_3, r_{n-2}) and so on. We can obtain the same result as that of $2n+1$ rows and columns by excluding the median row and column.

Rack Density

To provide an idea of the deployment density of Diamond, we give an example. A room of data center with the size $100\times100m^2$ can hold 1.98k racks if using Diamond, and hold 3.7k racks if using the conventional row-based architecture, so the density of the conventional architecture is about 1.8 times that of Diamond. When considering the bucket-based Diamond architecture, for the same room size with four buckets, it can hold 1.92k racks totally, which shows similar rack density as the single-bucket case. The lower rack density in Diamond ensures a proper space for both the wireless transmissions and cooling when the network scales up. However, our simulation results with different room sizes of a data center show that, the server-to-server throughput in Diamond on average doubles that of a conventional three-layer fat-tree DCN topology for the same room size [27].

Cabling Complexity

The cabling complexity is an important issue to consider in the deployment of DCNs. Despite their contributions to big performance improvements, both the tree-like topologies [27] and recursion-based topologies [28], [30] introduce complex cabling among racks and thus high maintenance cost in practice. This is because the physical row-by-row rack deployment does not work well with their logical tree or recursive topologies. In contrast, the cabling in Diamond is much easier with its wired structure simplified to be several rows and columns both logically and physically. As shown in Fig. 2(a) and Fig. 9, the row lines and column lines are independent from each other and thus are simple for both cabling and maintenance.

Cooling and Maintenance

Heat dissipation is important for a data center to run healthily. In conventional DCN architectures, the most challenging heat issue comes from the closely placed racks in multiple rows. Since the rack density in Diamond is both lower and more balanced (i.e., the distance between any two neighboring racks is similar) than conventional architectures, the heat is distributed more evenly and lightly. For better cooling effect in Diamond, we suggest piping the cooling air from bottom to top in each ring. In addition, we suggest leaving four gaps at the polygon corners evenly on each ring to form four tunnels through the innermost to the outermost, through which the engineers can go inside each ring for device maintenance. When there is human movement inside a data center, some wireless links may be blocked and fail. However, Diamond

(a) (b) (c)

Fig. 11. 60GHz antenna testbed for Diamond. (a) Transmit control panel. (b) Direct communication. (c) Multiple reflection.

TABLE I
TOTAL COST OF DIFFERENT DCN ARCHITECTURES

Topology #	Cost (k$)					Power (kw)
	NIC	Switch	Radio	Cabling	Total	
FatTree	80	2080	-	80	2240	3486
3DB	80	2080	192	80	2432	3486
FireFly	80	416	2400	16	2912	4281
Diamond	240	832	1920	32	3024	3428

can handle the failures of wireless links easily (Section IV-C) and quickly redirect data flows to alternative links and paths.

Moreover, the antenna steering delay may be an issue to affect the system performance. The delay of steering 60GHz antenna can potentially be controlled within 250us if using phase array technology [9], while if deploying FSO in Diamond, the steering delay can be within 0.5ms using Galvo mirrors [11]. To further alleviate the side effect, our system ensures that the transmissions through wireless links during the antenna steering to be easily migrated to the stable wired links.

Deployment Cost

A set of hybrid DCNs are proposed recently, such as the 3D-beamforming (3DB) [13] (8 radios per rack) and FireFly [11]. We use Fat-tree to represent the conventional wired architecture and compare the cost of different architectures in Table. I. We consider the cost and power of NICs on the server, switches, wireless radios and wires. We conservatively estimate each wireless radio costs $60 [12], each 40-port switch costs $1040, each port in the NIC costs $5 and needs 5W [28], each port in the FSO device costs $150 [11], and an average cost of $1 per meter for cabling [11] and $1 per square meter of absorbing paper. We assume the reflectors used in Firefly, 3DB and Diamond have negligible cost. The extended Diamond with multiple buckets has the same total cost of NICs, switches and radios as that of the normal single-bucket case, except that it has a cabling cost within twice that of the single-bucket case. All the architectures hold 16 thousand servers. We can see that although Diamond uses a large number of radios, its cost is only 24% higher than that of 3DB because it uses 60% fewer switches. This trade-off is reasonable as a larger number of wireless links are enabled in Diamond than 3DB. Firefly can offer higher bandwidth at a higher deployment cost. However, the ceiling mirror it requires may not be applicable in most modern data centers. An alternative solution is to replace 60Ghz radios in Diamond with FSO devices, which will provide similar performance as Firefly without the need of deploying ceiling mirrors but at a higher deployment cost.

VII. IMPLEMENTATION

We implement a 60GHz testbed to evaluate the transmission performance of our architecture under different wireless communication conditions.

Experiment Setup

To demonstrate the feasibility of 60GHz wireless communication in our architecture, we build a testbed (Fig. 11a) to carry out the relevant experiments. The testbed was composed by Vubiq Networks Inc's commercial millimeter wave transceiver components, self-designed 60GHz Power Amplifier and AINFO Inc's 60GHz rectangular waveguide horn antenna. The system enables 60 GHz experiments on the use of integrated transmitter/receiver waveguide modules. 60GHz Power Amplifier is placed at the end of the transmitter to increase the transmission power. It has a gain of 30dB and a saturated output power of 0.37W. The testbed encodes the data file with LPDC and applies the QPSK modulation to generate the waveform. The receiver module samples the signal and recovers the original data file.

We first carry out four experiments, including the direct communication, communication through single reflection, communication through double reflections and communication through deflection (i.e., the misalignment of two communicating antennas). In this group of experiments, to ensure the transmission ability of the architecture, the distance between the sender radio and the receiver radio is set to 25 m. The communication rate is 2.5 Gbps and the LPDC encoding rate is 3/4. We show the results in Fig. 12. For the second group of experiments, we change the hole size to test the performance of precise reflection for both the single and double reflection cases. To make an accurate measurement of hole size, the distance between the sender and receiver is set to 3m. The results are presented in Fig. 13 and Fig. 14.

Experiment Result on Signal Reflection

As Fig. 12 shows, the direct communication and the communications through single reflection and double reflections present a good communication quality and the corresponding SNR are 16.194 dB, 15.23 dB and 14.80 dB respectively. For all the experiments on our testbed, the measured data rates of both the directional and reflectional 60GHz links are shown to keep a value above 2.5Gbps over a distance of 25m. Therefore, the bandwidth of 60GHz wireless link is high enough for multiple-gigabit data transmissions in Diamond.

Experiment Result on Receiver Alignment

During the measurement, we find that the communication quality through deflection changes with the deflection angle between two radios. As Fig. 12d shows, when the deviation angle becomes 20°, the SNR is 12.75 dB, which is the critical value of the communication quality. When the deviation angle further increases, the communication quality becomes too bad for the receiver to decode the original data. This indicates that our 60GHz radio is highly directional and has a small main-lobe width less than 20°, which contributes to a small angular interference to other radios when constructing the wireless interference graph. At the same time, the main-lobe angle

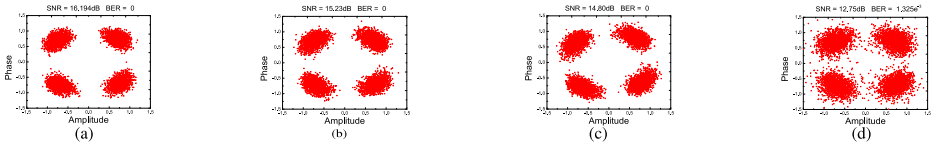


Fig. 12. Measured constellation diagram: performance of different transmission ways. (a) Direct communication. (b) Single reflection. (c) Double reflection. (d) Deflection.

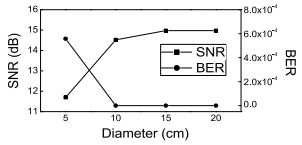


Fig. 13. Performance over different hole sizes.

provides a certain degree of fault tolerance on the antenna alignment between two servers in Diamond. We studied the impact of antenna misalignment through simulation with the above experimental parameters as input, and our result show that the average flow throughout drop is within 10% when the misaligned degree is within $\pm 20°$, which demonstrates that Diamond has a good tolerance to the fault as a result of the misalignment of antennas.

Experiment Result on the Precise Reflection

We examine the impact of hole size on the reflector and show the single-reflection performance in Fig.13. We are not showing the results with hole size larger than 20cm, because they are the same as the 20cm case. We can see that when the hole size is 10cm, the SNR gets a slight decrease but BER is kept at zero. When the hole size further decreases to 5cm, the SNR drops quickly and results in the transmission failure.

After obtaining the proper hole size as 10cm, we measure the constellation diagram for both the single and double reflections. Fig. 14(a) and Fig. 14(c) show the results of reflections without any absorbing materials on the reflector. Fig. 14(b) and Fig. 14(d) show the corresponding results with one 10cm x10cm hole on each reflector. We can see that the transmission performance keeps nearly the same for both cases. Another interesting finding is that for double reflections, the SNR even gets slightly better when the reflectors are full of absorbing paper with only one hole left. The gain may be achieved as a result of the reduction of the multiple-path interference with the use of absorbing material. This demonstrates the feasibility of using precise reflection in Diamond.

VIII. Simulation

A. Setup and Workloads

Our simulations are performed by a customized flow-level simulator. We use the same settings of TCP for the flow-level simulator as that utilized in [22], where the additive increase factor of flow rate is set to 15 MB/s. The wireless transmission follows the general physical interference and path loss model [18]. The related wireless parameters, such as the signal fading due to the misalignment of antennas, are all set following the testbed-based measurement results shown in Section VII.

For comparative analysis, we consider two classes of typical DCN topologies respectively: (1) wired topology and (2) hybrid topology. In the first part, we evaluate the performance of the wired backbone of Diamond (named Diamond-Wired) and other typical wired DCN topologies. The wired link capacity is set to 1Gbps, and we use Fat-tree [27] and BCube [28] as the representatives for the tree-based DCN topology and the recursion-based DCN topology respectively. In the second part, we evaluate the performance of Diamond and the state-of-art hybrid architecture 3D-beamforming [13]. We apply Fat-tree as the oversubscribed core for 3D-beamforming. Since 3D-beamforming deploys the wireless radios only at the ToR layer, to make a fair comparison, we apply two radios on top of each rack for both 3D-beamforming and Diamond. Thus, only the first layer of servers in Diamond are equipped with wireless radios and the radio numbers are the same for both topologies. To compare the performance only under distributed routing, we further disable the HDF (Highest Demand First algorithm) function and only use the OHR (Opportunistic Hybrid Routing) routing in Diamond (named Diamond-OHR), while 3D-beamforming uses ECMP routing [42]. Limited by the memory space of our simulator, the number of rings in Diamond is set to six.

TABLE II
Evaluation Parameters Setup

	Default	Range
Average Flow Size (MB)	100	0 $\sim$ 500
The degree of hotspot traffic (%)	0	0 $\sim$ 50
The degree of node failure (%)	0	5 $\sim$ 25
Max wireless distance (m)	10	3 $\sim$ 10
Bucket number	1	1 $\sim$ 4

To compare the performance on load balancing and fault tolerance, we evaluate Diamond and other DCN topologies under different traffic patterns and number of node failures. The HDF routing and wireless radios are all enabled for Diamond (named Diamond-HDF) in the comparison cases. We transfer 200 random flows with their sizes set within 200MB, and show the performance results of flow completion time and throughput.

We evaluate the flow performance for the bucket-based Diamond architecture using different number of bucket components. We intend to illustrate the low wireless efficiency when the wireless can only cover a small fraction of transmissions in the rings when the diameters are very large. Unfortunately, we can not simulate Diamond with very large rings in the normal setting due to the memory limit of our simulator. Thus we propose to approximate this case by reducing the maximum wireless transmission distance in simulations. To make a fair comparison, the compared architectures are holding the same number of racks: six rings if using only one bucket, four rings per bucket if using two buckets and three rings per bucket if using four buckets. For clarity, we list the evaluation parameter setup in Table II.

B. Performance of Wired and Hybrid Architecture

Wired Arhictecture: In Fig. 15a, we can see that BCube performs the best while Diamond-Wired has similar flow

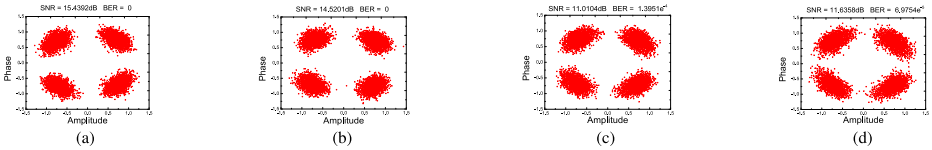


Fig. 14. Measured constellation diagram: performance of precise reflection. (a) Single reflection without absorbing. (b) Single reflection on one 10cm × 10cm hole. (c) Double reflections without absorbing. (d) Double reflections on two 10cm × 10cm holes.

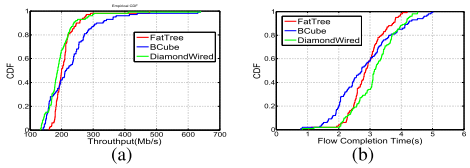


Fig. 15. Flow performance of wired architectures. (a) Flow throughput. (b) Flow completion time.

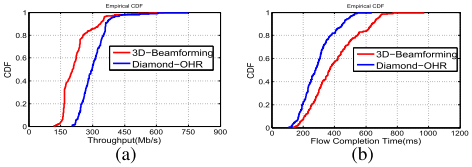


Fig. 16. Flow performance of hybrid architectures. (a) Throughput (long flows). (b) Completion time (short flows).

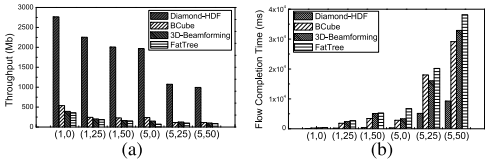


Fig. 17. Flow performance of different traffic patterns. (a) Flow throughput. (b) Flow completion time.

throughput as Fat-tree. The number of flows whose throughput is larger than 300Mbps takes 10% in BCube, while the percentage is less than 1% in the other two topologies. This is because that DiamondWired simplifies its wired backbone by using much fewer switches and wires. Similar trends on the performance of flow completion time can be found in Fig. 15b.

Hybrid Architecture: Consider the original traffic as long flows. We add another 200 random short flows (whose average size is one tenth that of the original traffic) to study the performance of mixed flows in hybrid architectures. In Fig. 16a, the throughput of long flows in Diamond-OHR is higher than that of 3D-beamforming. The number of long flows whose throughput is larger than 225Mbps takes above 90% in Diamond, while the number takes less than 40% in 3D-beamforming. Moreover, in Fig. 16b, the maximum completion time of short flows in Diamond is about 25% less than that of 3D-beamforming. In Diamond, a larger number of concurrent wireless links can be supported to increase the transmission capacity, which contributes to both higher throughput for long flows and smaller completion time for short flows.

C. Performance of Load Balancing

Following the prior work [11], we use a uniform model where flows between pairs of racks arrive independently with a Poisson arrival-rate as the baseline. We also consider the hotspot model [23], where in addition to the uniform baseline, a subset of rack pairs have higher arrival rates and larger flow sizes. We use a tuple (X, Y) to describe the hotspot traffic model: the X element represents the average flow size, where 1 denotes the average flow size is 100MB, and 5 corresponds to 500MB; the Y element denotes the percentage of the number of hot nodes.

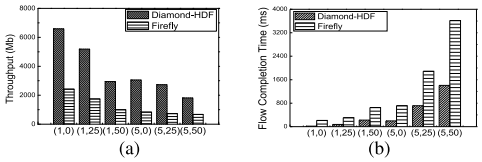


Fig. 18. Flow performance for FSO wireless. (a) Flow throughput. (b) Flow completion time.

As Fig. 17 shows, the flow performance of the four topologies deteriorates as expected when increasing the average flow size and the number of hot nodes. Diamond-HDF performs the best, providing the largest flow throughput and lowest flow completion time. Benefited from the rich server-level wireless links, the throughput of Diamond is about 5 times that of other topologies in the lightest traffic case $(1, 0)$, and 9 times that of the other topologies in the worst traffic case $(5, 50)$. Correspondingly, the flow completion time of Diamond is about 70% lower than that of other topologies. This demonstrates the high performance gains of Diamond-HDF and its capability of effectively balancing the load upon heavy traffic.

In Fig. 18, we study the flow performance when using the high-performance FSO wireless devices, which is proposed in recent work Firefly [11], to replace 60GHz wireless radios in Diamond. Since the FSO device can achieve a wireless bandwidth above 10Gbps, to make a reasonable comparison, the wired link capacity is also updated to 10Gbps in the simulation. We can see that the throughput of Diamond still about twice that of Firefly, and the flow completion time of Diamond is about 60% lower than that of Firefly in the the worst traffic case $(5, 50)$. This is because Firefly only uses limited rack-level wireless links, while Diamond can exploit a large number of server-level high-performance FSO wireless links for better load balancing thus higher performance than that of Firefly.

D. Performance of Fault Tolerance

In Fig. 19, we evaluate the flow performance of Diamond-HDF and Diamond-Wired when different percentages of nodes fail. To ensure that every flow can be routed under the node failures, we first randomly disable certain percentage of nodes and then randomly generate 100 flows to transmit for the remaining nodes. As Fig. 19 shows, the flow throughput of both the Diamond-HDF and Diamond-Wired decreases

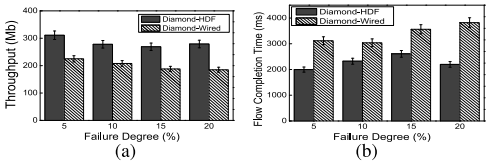


Fig. 19. Flow performance for fault tolerance. (a) Flow throughput. (b) Flow completion time.

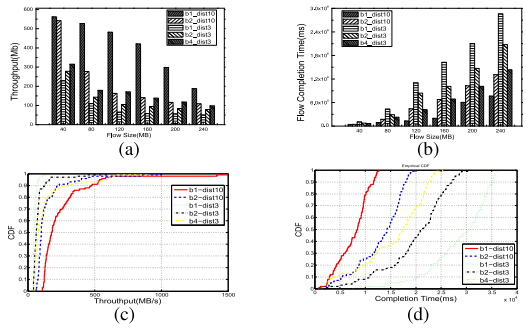


Fig. 20. Flow performance of bucket-based architecture. (a) Flow throughput. (b) Flow completion time. (c) CDF of flow throughput. (d) CDF of flow completion time.

with the increasing node failure ratio. However, the flow throughout of Diamond-HDF decreases much slower than that of Diamond-Wired. Considering the failure ratio from 0% to 20%, the flow throughput of Diamond-HDF decreases about 13% while Diamond-Wired decreases about 28%. This illustrates the graceful performance degradation of Diamond-HDF for node failures. Similar trends on flow completion time can be found in Fig. 19b.

E. Performance of Bucket-Based Diamond Architecture

In Fig. 20, we evaluate the performance of the bucket-based Diamond architecture. The baseline case is *b1-dist10* which has one bucket and the maximum wireless transmission distance is 10m. We can see that *b1-dist10* gains highest throughput and lowest completion time among all the cases. With this distance, we find that using more buckets (*b2-dist10*) will degrade the performance, as such a distance already covers enough fraction of the six-ring bucket to achieve a high wireless efficiency. When using more buckets with fewer rings in each, some high-performance wireless links are removed while more wired links are added to connect different buckets. Thus with enough wireless efficiency in one bucket, it is better using a single bucket rather than multiple buckets. However, when the maximum wireless transmission distance is reduced to 3m, the *b1-dist3* gets the worst performance among all the cases, which demonstrates that a low wireless efficiency will seriously reduce the whole transmission performance. To avoid the performance degradation, both *b2-dist3* and *b4-dist3* use more buckets with fewer rings in each. We can see the average flow completion time is reduced by about 30% and 50% respectively. With a small number of rings in each bucket, the relative ratio between wireless transmission distance and ring size is kept high for a guaranteed wireless efficiency and transmission performance.

F. Performance of Wireless Reconfiguration

In Table III, we compare the computation delay and performance of the greedy solution HDF in Diamond with

TABLE III
PERFORMANCE OF RECONFIGURATION

Ring #	Delay (ms)		Throughput Gap	Flow Completion Time Gap
	Full-ILP	HDF		
2	219	15	0.08	0.11
3	313	31	0.08	0.15
4	625	31	0.12	0.01
5	11625	32	0.15	0.15

the optimal solution (named Full-ILP) of the HLB problem. We use the ILP solver LINGO to compute the global optimal solution of ILP for routing (we obtained the same results when using the ILP toolbox in MATLAB for calculation). Limited by the memory constraint of LINGO, we evaluate the scales of Diamond with up to 5 rings which contains totally 60 racks and each rack holds 48 servers. We can see that the computation delay of Full-ILP increases quickly with the number of rings while HDF keeps a stable and low computation delay around 30ms. The tradeoff is HDF gets up to 15% gap on the performance of throughput and flow completion time when compared with Full-ILP. For a practical network scale within 20 rings, Full-ILP can not provide the solution in reasonable time, while HDF still achieves a low delay within 100ms, which is comparable to the feasible scheduling overhead illustrated in [22].

IX. RELATED WORK

Conventional Data Center

There exist prevalent hot spots in hierarchical data centers [6], [9], [12], [27], which limits the DCN performance. Many DCN architectures have been proposed to address the hot-spot problem in tree-based data center networks. Some efforts [26], [38], [43] propose to construct a random networking topology to achieve smaller network diameter, less hot spots and higher performance than state-of-art structured architectures. But the wiring and routing are quite challenging in a totally random wired network. Guo *et al.* [28], [30] propose to build the network recursively to efficiently eliminate the structured bottleneck. However, the routing is restricted to follow its recursive structure, which does not consider the high dynamics in traffic demands and thus may lead to more hot spots.

Hybrid Data Center Networking

Recent efforts turn to hybrid data center networking with flexible new networking components (e.g., the optical circuit switches, 60GHz wireless radios or FSO transceivers) to address the dynamic traffic demands. c-Through [8] enables flexible OCS links among all the ToR switches, while Helios [7] focuses the OCS links among different pod switches. Rather than restricting the topology flexibility by the single-hop optical links, OSA [44] proposes an architecture that supports higher topology flexibility via multi-hop optical links. Besides, xFabric [45] uses an architecture where the OCS links in racks can also be reconfigured. Recently, with the help of ToR buffer, RotorNet [20] achieves global flexibility by cycling through a series of optical matchings. Flat-tree [46] could change the network to one of the three predefined architectures with small "converter swtiches". Flyway first illustrates the feasibility of applying 60GHz wireless technology in DCNs [9]. The work in [13] further enhances the Flyway performance by using the ceiling reflector to

bounce signals to avoid blocking on the 2D plane. Using the same method, Firefly explores the feasibility of running free-space-optical (FSO) transmissions in DCNs [11]. This method, however, requires a height-restricted ceiling and also complete clearance above racks, which is infeasible in most data centers due to the existence of air conditioning pipes and steel structures above the racks [47]. ProjecToR enables direct links between all pairs of racks with free-space optics [14], while it faces many challenges in practical data center environments [19]. Moreover, existing methods only considered the local performance improvement at the rack level and part of network layers. In contrast, Diamond can run a larger number of network-wide wireless links (either 60GHz or FSO) without involving any engineering efforts to change the room plan above racks. Both wireless technologies can be applied in Diamond at the server level with different trade-offs: commodity 60GHz antenna is much cheaper and smaller than FSO transceivers while FSO has little interference footprint and longer transmission distance. With the decreasing cost of optical transceivers, FSO shows great promise to run in Diamond in the future.

X. Conclusion

We propose Diamond, a novel hybrid network architecture, to enable high capacity and seamless data transmissions over both wired and wireless network links. Specifically, we introduce the concept of Ring Reflection Space (RRS) to enable the wide deployment of wireless radios at servers and high number of concurrent wireless transmissions through low-cost multi-reflection over the metal, and develop a precise reflection scheme to reduce the wireless interference inside an RRS. The rich wireless resources allow Diamond to flexibly configure the network topology and form the transmission path to avoid creating hot traffic spots while enabling transmissions over random network topology for low delay. We also design a modular component termed the bucket to achieve a fully-scalable Diamond architecture. We implement the proposed techniques over 60Ghz testbed and demonstrate its functionality. Our results from extensive simulations show that the cohesive structure of Diamond enables fine-grained and network-wide load balancing, effective routing and graceful fault-tolerance.

References

[1] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements & analysis," in *Proc. SIGCOMM*, 2009, pp. 202–208.

[2] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 92–99, 2010.

[3] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proc. IMC*, 2010, pp. 267–280.

[4] M. Alizadeh *et al.*, "Data center TCP (DCTCP)," in *Proc. SIGCOMM*, 2011, pp. 63–74.

[5] L. Xu, K. Xu, Y. Jiang, F. Ren, and H. Wang, "Throughput optimization of TCP incast congestion control in large-scale datacenter networks," *Comput. Netw., Int. J. Comput. Telecommun. Netw.*, vol. 24, pp. 46–60, Sep. 2017.

[6] A. Greenberg *et al.*, "Vl2: A scalable and flexible data center network," in *Proc. SIGCOMM*, 2009, pp. 51–62.

[7] N. Farrington *et al.*, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. SIGCOMM*, 2011, pp. 339–350.

[8] G. Wang *et al.*, "c-Through: Part-time optics in data centers," in *Proc. SIGCOMM*, 2010, pp. 327–338.

[9] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, "Augmenting data center networks with multi-gigabit wireless links," in *Proc. SIGCOMM*, 2011, pp. 38–49.

[10] M. Yu *et al.*, "Profiling network performance for multi-tier data center applications," in *Proc. NSDI*, 2011, pp. 57–70.

[11] N. Hamedazimi *et al.*, "FireFly: A reconfigurable wireless data center fabric using free-space optics," in *Proc. SIGCOMM*, 2014, pp. 319–330.

[12] Y. Zhu *et al.*, "Cutting the cord: A robust wireless facilities network for data centers," in *Proc. MobiCom*, 2014, pp. 581–592.

[13] X. Zhou *et al.*, "Mirror mirror on the ceiling: Flexible wireless links for data centers," in *Proc. SIGCOMM*, 2012, pp. 443–454.

[14] M. Ghobadi *et al.*, "ProjecToR: Agile reconfigurable data center interconnect," in *Proc. SIGCOMM*, 2016, pp. 216–229.

[15] H. Liu *et al.*, "Scheduling techniques for hybrid circuit/packet networks," in *Proc. CoNEXT*, 2015, Art. no. 41.

[16] K. Chen *et al.*, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 498–511, Apr. 2014.

[17] G. Porter *et al.*, "Integrating microsecond circuit switching into the data center," in *Proc. SIGCOMM*, 2013, pp. 447–458.

[18] Y. Cui, H. Wang, X. Cheng, D. Li, and A. Ylä-Jääski, "Dynamic scheduling for wireless data center networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 12, pp. 2365–2374, Dec. 2013.

[19] L. Chen *et al.*, "Enabling wide-spread communications on optical fabric with MegaSwitch," in *Proc. NSDI*, 2017, pp. 577–593.

[20] W. M. Mellette *et al.*, "RotorNet: A scalable, low-complexity, optical datacenter network," in *Proc. ACM SIGCOMM*, 2017, pp. 267–280.

[21] J.-Y. Shin, E. G. Sirer, H. Weatherspoon, and D. Kirovski, "On the feasibility of completely wireless datacenters," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1666–1679, Oct. 2013.

[22] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, 2010, p. 19.

[23] A. R. Curtis *et al.*, "DevoFlow: Scaling flow management for high-performance networks," in *Proc. SIGCOMM*, 2011, pp. 254–265.

[24] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, "Fastpass: A centralized 'zero-queue' datacenter network," in *Proc. SIGCOMM*, 2014, pp. 307–318.

[25] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[26] A. Singla, P. B. Godfrey, and A. Kolla, "High throughput data center topology design," in *Proc. NSDI*, 2014, pp. 29–41.

[27] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. SIGCOMM*, 2008, pp. 63–74.

[28] C. Guo *et al.*, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. SIGCOMM*, 2009, pp. 63–74.

[29] N. G. de Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie Wetenschappen*, vol. 49, no. 7, pp. 758–764, Jun. 1946.

[30] C. Guo *et al.*, "DCell: A scalable and fault-tolerant network structure for data centers," in *Proc. SIGCOMM*, 2008, pp. 75–86.

[31] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979, pp. 61–62.

[32] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proc. MobiCom*, 2006, pp. 262–273.

[33] X.-Y. Li, S.-J. Tang, and O. Frieder, "Multicast capacity for large scale wireless ad hoc networks," in *Proc. MobiCom*, 2007, pp. 266–277.

[34] X.-Y. Li and Y. Wang, "Simple approximation algorithms and PTASs for various problems in wireless ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 66, no. 4, pp. 515–530, 2006.

[35] Y. Wang, W. Wang, X.-Y. Li, and W.-Z. Song, "Interference-aware joint routing and TDMA link scheduling for static wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 12, pp. 1709–1726, Dec. 2008.

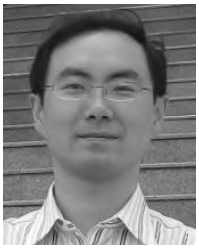
[36] X. Y. Li, "Multicast capacity of wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 950–961, Jun. 2009.

[37] X. Xu, X.-Y. Li, P.-J. Wan, and S. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 690–698, Jun. 2012.

[38] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Proc. NSDI*, 2012, p. 17.

[39] A. Basu and J. Riecke, "Stability issues in OSPF routing," in *Proc. SIGCOMM*, 2001, pp. 225–236.

[40] C. Camarero, E. Vallejo, and R. Beivide, "Topological characterization of Hamming and dragonfly networks and its implications on routing," *ACM Trans. Archit. Code Optim.*, vol. 11, no. 4, 2015, Art. no. 39.
[41] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, and A. Donnelly, "Symbiotic routing in future data centers," in *Proc. SIGCOMM*, 2011, pp. 51–62.
[42] K. Xu, M. Shen, H. Liu, J. Liu, F. Li, and T. Li, "Achieving optimal traffic engineering using a generalized routing framework," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 51–65, Jan. 2016.
[43] J.-Y. Shin, B. Wong, and E. G. Sirer, "Small-world datacenters," in *Proc. 2nd ACM Symp. Cloud Comput.*, 2011, Art. no. 2.
[44] K. Chen *et al.*, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," in *Proc. NSDI*, 2012, p. 18.
[45] S. Legtchenko *et al.*, "XFabric: A reconfigurable in-rack network for rack-scale computers," in *Proc. NSDI*, 2016, pp. 15–29.
[46] Y. Xia *et al.*, "A tale of two topologies: Exploring convertible data center network architectures with flat-tree," in *Proc. SIGCOMM*, 2017, pp. 295–308.
[47] *Google Data Center*. [Online]. Available: http://www.google.com/about/datacenters/gallery

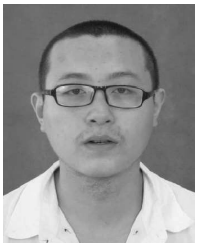


Yong Cui received the B.E. and Ph.D. degrees in computer science and engineering from Tsinghua University, China, in 1999 and 2004, respectively. He is currently a Full Professor with the Computer Science Department, Tsinghua University. He has authored over 100 papers in refereed conferences and journals with several Best Paper Awards. He has coauthored seven Internet standard documents (RFC) for his proposal on IPv6 technologies. His major research interests include mobile cloud computing and network architecture. He served or serves on the Editorial Boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and the *IEEE Internet Computing*. He is currently the Working Group Co-Chair in the IETF.

Shihan Xiao received the B.Eng. degree in electronic and information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing. His research interests are in the areas of data center networking and cloud computing.

Xin Wang received the B.S. and M.S. degrees in telecommunications engineering and wireless communications engineering from the Beijing University of Posts and Telecommunications, Beijing, China, and the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY, USA. She was a member of Technical Staff in the area of mobile and wireless networking with Bell Labs Research, Lucent Technologies, NJ, USA, and an Assistant Professor with the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, USA. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, State University of New York, Stony Brook, NY, USA. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, and networked sensing and detection. She received the NSF Career Award in 2005 and the ONR Challenge Award in 2010. She has served for the Executive Committee and Technical Committee of numerous conferences and funding review panels and serves as an Associate Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING.

Zhenjie Yang received the B.E. degree in networking engineering from the Dalian University of Technology, Liaoning, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include data center networking and cloud computing.



Shenghui Yan received the B.S. degree from the Computer Science and Technology Department, Beihang University, Beijing, China, in 2014, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, in 2017. His research interests are in the areas of data center networking.

Chao Zhu received the B.E. degree in computer science and technology from the Beijing University of Posts and Telecommunications, China, in 2012, and the M.S. degree in computer technology from Tsinghua University, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Science, Aalto University, Finland. His research interests include mobile computing and vehicular networking.

Xiang-Yang Li (M'00–SM'08–F'15) received the bachelor's degree from the Department of Computer Science and the Department of Business Management, Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, 2000 and 2001, respectively. He was a Full Professor with the Illinois Institute of Technology, Chicago, IL, USA. He is currently a Full Professor and an Executive Dean with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China. He is an ACM Distinguished Scientist. He has authored the monograph *Wireless Ad Hoc and Sensor Networks: Theory and Applications.*

Ning Ge (M'01) received the B.S. and Ph.D. degrees from Tsinghua University, China, in 1993 and 1997, respectively. From 1998 to 2000, he was with ADC Telecommunications, Dallas, TX, USA, involving in the development of ATM switch fabric ASIC. Since 2000, he has been with the Department of Electronics Engineering, Tsinghua University, where he is currently a Professor and serves as the Director of the Communication Institute. His research interests include ASIC design, short-range wireless communication, and wireless communications. He is a Senior Member of CIC and CIE.