Providing Line-of-Sight in a Free-Space-Optics Based Data Center Architecture

Max Curran and Himanshu Gupta
Stony Brook University, Stony Brook, NY 11794

ABSTRACT

To overcome the shortcomings of traditional static (wired) data center architectures, we recently proposed FireFly, a fully-flexible and fully-wireless data center network fabric based on free space optical (FSO) links. To facilitate a clear line-of-sight between the FSO devices placed on racks, FireFly uses a full-ceiling mirror for beam redirection. Use of a full-ceiling mirror imposes significant operational and infrastructural challenge and expense. The focus of our paper is to propose and evaluate alternative schemes to provide line-of-sight for FSO links in FireFly. In particular, we propose two schemes: (i) strategically placed multiple but small overhead mirrors, and (2) "towers" on top of racks, on which FSOs can be placed at regular heights. We develop comprehensive techniques for the above suggested schemes, and demonstrate the viability of these schemes by evaluating their performance using simulations for various performance metrics of interest.

1 Introduction

Data centers (DCs) are a critical piece of today's networked applications in both private and public sectors (e.g., [2, 3, 6–8]). A robust *datacenter network fabric* is fundamental to the success of DCs and to ensure that the network does not become a bottleneck for high-performance applications [19]. In this context, DC network design must satisfy several goals: high performance [10, 16], low equipment and management cost [10, 25], robustness to dynamic traffic patterns [17, 26, 28, 29], incremental expandability to add new servers or racks [14, 27], and other practical concerns such as cabling complexity [23], and power and cooling costs [15, 24].

Traditional data center architectures have been based on wired networks; being *static* in nature, these networks have either been (i) *overprovisioned* to account for worst-case traffic patterns, and thus incur high cost (e.g., fat-trees or Clos [10, 13, 16]), or (ii) *oversubscribed* (e.g., simple trees or leaf-spine architectures [1]) which incur low cost but offer poor performance due to congested links. Recent works have tried to overcome the above limitations by augmenting a static (wired) "core" with some flexible links (RF-wireless [17, 29] or optical [12, 28]). These *augmented* architecture show promise, but have offered only incremental improvement in performance due to various limiting factors. Furthermore, all the above architectures incur high cabling cost and complexity [23].

To overcome the above cost-performance tradeoffs and rigidity of past DC architectures, in our recent work [18], we proposed an *extreme* design point—a *fully flexible*, *all-wireless* inter-rack fabric using *Free-Space Optics (FSO)* communication links. The FSO communication technology is particularly well-suited as it can offer very high data rates (tens of Gbps) over long ranges (>100m) using low transmission power and with small interference footprint [21]. Fig. 1 shows a conceptual overview of our proposed design called *FireFly*. A number of FSO devices are placed on top of each rack and are connected to the top-of-the-rack switch. Each FSO device assembly is capable of precise and fast steering to connect to FSO devices on other racks. The controller intelligently reconfigures these devices in real-time to adapt to changing network requirements. Since the FSO beams may be obstructed by other devices in the system, FireFly proposes use of a ceiling mirror for beam redirection to ensure clear line-of-sight.

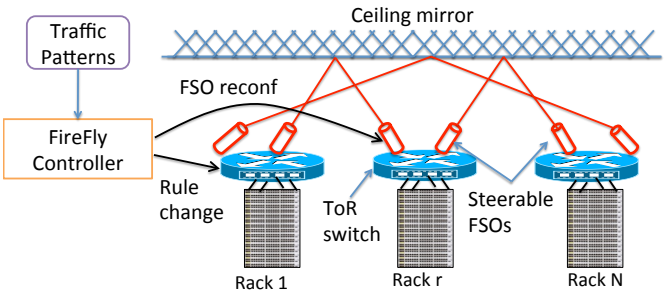

Figure 1: High-level view of FireFly.

Paper Contribution and Organization. One of the key shortcomings of the FireFly architecture is the proposed use of a full ceiling mirror. A full ceiling mirror poses a significant operational and infrastructural challenge and expense. In this paper, we propose two alternatives to having a full ceiling mirror, viz., small overhead mirrors (§3) and towers (§4) on racks, and develop comprehensive techniques for each of these techniques. Through extensive simulations (§5), we demonstrate the viability of the proposed alternatives. We start with a background on FireFly in §2.

2 Background: FireFly Overview

In this section, we give an overview of the FireFly architecture and introduce a couple of terms viz., dynamic graphs and candidate links.

As mentioned in the previous section, FireFly is a fully-flexible all-wireless data center network architecture. It is based on the key insight that flexibility can facilitate near-optimal performance when done right. Below, we discuss FireFly's key components and benefits, so as to provide the necessary background and context.

FireFly Components. The FireFly architecture for data centers comprise of following key components, viz., the FSO devices, link steering mechanisms, and the network management techniques.

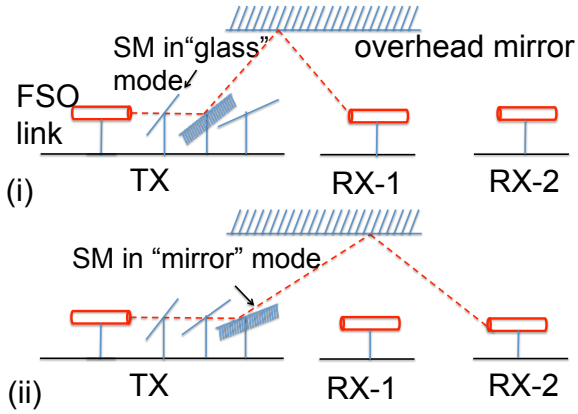

Figure 2: Using SMs to steer the TX-beam from (i) RX-1 to (ii) RX-2. RX-s also have similar setup of SMs (not shown).

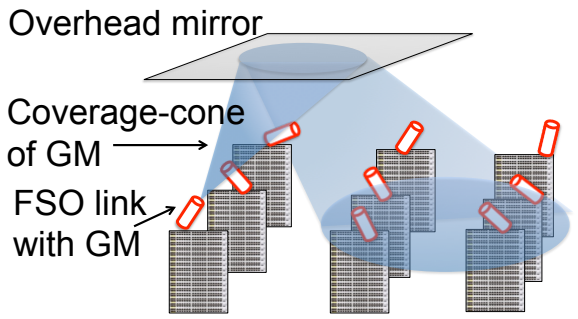

Figure 3: Movable mirror (GM) on an FSO device can steer the beam within its coverage-cone.

FSO devices. FSOs are needed to create FSO communication links in FireFly need to have a small form factor (so a few tens of them can fit on the top of a rack), and must be able to deliver high data rates at distance of 100 meters or more. and have low enough power consumption. In our prior work [18], we demonstrated a design of an FSO link prototype based on SFPs that satisfies the desired requirements.

Link Steering Mechanisms. For network reconfigurability, the FSO devices in FireFly are equipped with a mechanism to steer the laser beam from one receiver to the next; for viable performance, this steering must incur very low latency, i.e., in order of a few milliseconds. In [18], two types of steering mechanisms, viz., switchable mirrors (SMs) and Galvo mirrors (GMs), were explored and their feasibility for FireFly demonstrated. SMs are made from a special liquid crystal material that can be electrically controlled to rapidly switch between reflection (mirror) and transparent (glass) states at millisecond timescales [5]. To use SMs for beamsteering, each FSO device is equipped with multiple SMs with each SM *pre-aligned* (done offline) to target a receiving FSO. The desired link is established by switching one of the SMs in the mirror state, while leaving the other SMs in the transparent state [18]. See Fig. 2. Galvo mirrors (GMs) [4] use one or more small mirrors that move along specific axes within an angular range in response to a control signal. This in effect can redirect a reflected FSO beam within a pre-determined coverage cone (referred to as *pre-orientation* of the GM; chosen offline). See Fig. 3.

Network Management. Management of the FireFly network involves two key tasks: (i) Due to physical and geometric constraints, there is a need to *preconfigure* the network (done offline) with an appropriate number of FSOs devices, each equipped with a pre-oriented GM or pre-aligned SMs; this preconfiguration essentially creates a "dynamic graph" (see below) of possible links for use in real-time. (ii) Then, at runtime, the FireFly controller needs to dynamically selects a *runtime topology* and configure routing tables at each switch, based on prevailing traffic demands and events.

Cost, Performance, and Benefits. Our prior work [18] analyzed FireFly's performance and cost in comparison to several DC architectures. We estimated FireFly's cost to be much less than other architectures, and observed that, under simple traffic patterns, FireFly's performance was comparable to that of the full-bisection bandwidth networks and much better than the augmented architectures viz., 60-GHz based 3D-beamforming [29] and optical-based CThru [28]. We refer the reader to [18] for more details.

FireFly promises several benefits in comparison to current DC architectures. First, topological flexibility (if done right) can provides a low-cost solution (e.g., fewer switches, links) with near-optimal performance. Second, an all-wireless fabric eliminates the cabling complexity and associated overheads (e.g., obstructed cooling) [23]. Third, a fully-wireless architecture can facilitate new and radical DC topology structures that would otherwise remain "paper designs" due to cabling complexity [27]. Finally, flexibility enables easier incremental expansion of a DC and takes us closer to the vision of energy-proportional DCs [9, 11, 20] by allowing the flexibility to turn links on or off.

Line-of-Sight via a Full-Ceiling Mirror. FSO beams require a clear line-of-sight (LOS) between the transmitter and receiver for communication. However, beams emanating from the FSO devices placed on top of the racks are likely to be obstructed by other FSO devices. To circumvent these obstructions, [18] assumes a *full ceiling mirror* (as in prior related works [29]) for beam redirection. However, a full ceiling mirror introduces significant infrastructure and operational challenges, as confirmed by our discussions from DC operators. Thus, the focus of this paper is to explore alternate techniques to establish line of sight.

Background Terms: Dynamic Graph and Candidate Links.

Dynamic Graph. Consider a FireFly network, i.e., a set of racks each with a set of FSO devices each equipped with a pre-oriented GM or pre-aligned SMs. We can establish a *candidate (bi-directional) link* between a pair of FSOs a and b if (i) a has an SM aligned towards b and vice-versa or (ii) a is located in the coverage-cone of the GM at b and vice-versa. At any instant, only one candidate link per FSO can be an *active link*. For example, in Fig. 2, links (TX, RX-1) and (TX, RX-2) are candidate links, and link (TX, RX-1) is active in Fig. 2(i) while (TX, RX-2) is active in Fig. 2(ii). We refer to the set of all candidate links as the *dynamic graph*. Given a dynamic graph, we refer to a set of candidate links that can be active *simultaneously* as a *realizable (or runtime) topology*. See Fig. 4. Note that the only constraint on a set of links to be active simultaneously is that each FSO has at most one active candidate link incident on it, due to lack of wireless interference.

Unassigned and Assigned Candidate Links. The dynamic graph and realizable topologies can be viewed as graphs over the FSOs (with candidate/active links between pairs of FSOs), or over the racks (with candidate/active links between the corresponding racks). However, when a dynamic graph is looked upon as a graph over the racks, the information about the association of candidate links to individual FSOs is lost. In this paper, we would sometimes use the notion of dynamic graph over racks, and refer to the candidate links in such a dynamic graph as *unassigned candidate links* (in contrast, the candidate links in the dynamic graph over FSOs are *assigned* candidate links).

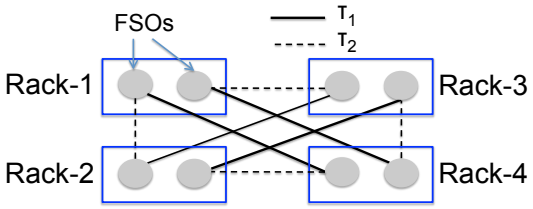

Figure 4: A dynamic graph with candidate links (solid and dashed). The set of solid links represents one possible realizable topology (τ_1), and the set of dashed lines represents another (τ_2).

3 Line-of-sight via Overhead Mirrors

In this section, we describe a strategy that facilitates line-of-sight (LOS) between FSO devices using a sufficient number of relatively small mirrors in lieu of a full ceiling mirror. We start with a definition and the key ideas behind our techniques, and then present our techniques for SM-based and GM-based FireFly architectures.

DEFINITION 1. (Reflection Point) Consider a FireFly network with a ceiling (i.e., the height at which horizontal overhead mirrors may be installed for beam redirection). The *reflection point* for an assigned candidate link connecting two FSOs a and b is the point p on the ceiling such that (i) a, b, and p are on the same vertical plane, and (ii) p is equidistant from $a$and b. Essentially, p is the point on the ceiling such that a beam from a when directed towards p on a horizontal mirror will reach b. □

Key Ideas. Our strategy to achieve LOS via placement of a number of small overhead mirrors is based on the following insights.

1. First, observe that, for any dynamic graph, we can provide LOS for each link individually and independently by placing a small mirror at its reflection point. These small mirrors can be very small, e.g., 1 cm × 1 cm to accommodate anticipated misalignment tolerance.

2. The above approach however is likely to result in too many mirrors – since the number of candidate links may be of the order of $O(n^2)$ where n is the number of racks. However, we observe that the "regularity" of the rack locations and the FSO positions on top of the racks, will result in many of these small mirrors to be located close-by. Thus, we can "combine" multiple small mirrors into a slightly-larger mirror, leading to a more manageable solution – comprising of a smaller number of slightly-larger mirrors. E.g., Figure 5 shows placement of less than $4n$ mirrors, each of size equal to the top of the rack, that are sufficient to provide LOS between FSOs placed on a rectangular grid of n racks.

3. We can further reduce the number and size of the individual mirrors, by a careful assignment of (unassigned) candidate links to FSOs. See Figure 6 for an illustration.

In the following subsections, we employ the above ideas to develop techniques that aim to provide LOS for FSO links using a sufficient number of small overhead mirrors, for each of the two FireFly architectures, viz., SM-based and GM-based.

3.1 SM-based DC Architecture

In this subsection, we address the problem of establishing LOS in FireFly with SMs as the steering mechanism. We start with the problem definition.

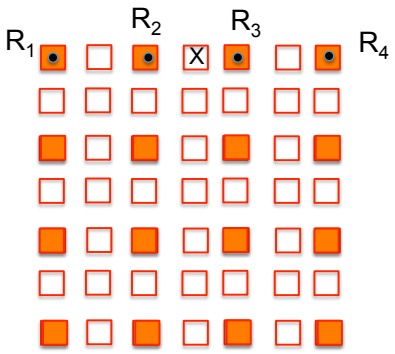

Figure 5: Top view of a DC with 16 racks (*solid* squares), and 49 overhead mirrors placed above all (solid and hollow) squares. The mirror marked "X" provides LOS between FSOs on racks R_1 and R_4 and R_2 and R_3.

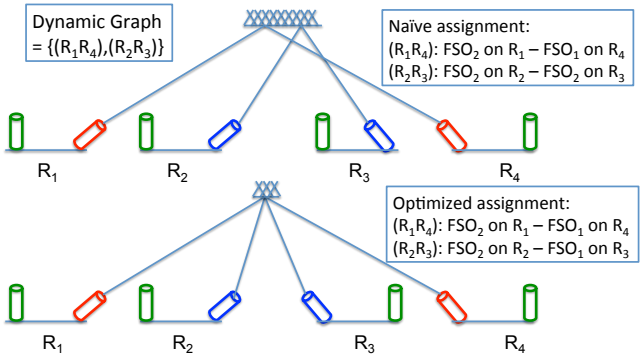

Figure 6: Careful assignment of links to FSOs can reduce the number of points and hence, the size of overhead mirrors required.

SMM Problem. Informally, the SMM problem is to create a dynamic graph with near-optimal performance over the given network configuration, while trying to minimize the usage of overhead mirrors. For simplicity, we represent the network performance by the number of candidate links created.

Formally, given the FireFly network parameters, i.e., layout of the racks, # of FSOs and their placement on each rack, # of SMs per FSO, the goal of the SMM problem is two-fold: (i) to create a dynamic graph over FSOs, and (ii) to determine the positions of overhead mirrors to place on the ceiling to provide LOS for the candidate links in the dynamic graph. The optimization objective is two-fold: (i) maximize the number of rack-pairs that have at least one candidate link between them in the dynamic graph,[1] and (ii) minimize the total area of *rectangular* overhead mirrors, under the constraint that the total number of mirrors used is below a given constant r.

SMM Algorithm. We start with creating a random simple graph G over racks such that each rack has mk links, where m is the number of FSOs per rack and k is the number of SMs per FSO (assuming mk is less than the total number of racks). We create a *random* graph, based on the insights from prior works [18, 27]. Note that G has the maximum number of links possible in any dynamic graph. Our algorithm tries to assign these (unassigned) candidate links of G to FSOs, and thus, creating a dynamic graph over the FSOs. Since the above method creates the maximum number of candidate links possible in a dynamic graph, we can now just focus

[1]This is slightly different than simply maximizing the total number of candidate links, since some rack-pairs may have multiple candidate links between them.

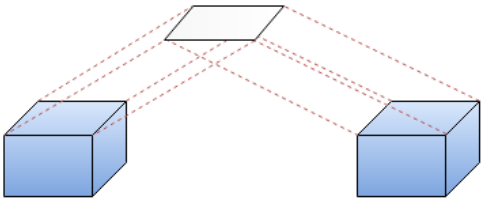

Figure 7: A pair of racks and its associated reflection polygon.

on minimizing the total area of the mirrors placed.

The basic idea of our algorithm is to assign links to FSOs in a way so that the resulting reflection points of the assigned candidate links can be clustered (based on location) into a small number of clusters (ideally, into r clusters). We achieve the above in two steps. First, we cluster the unassigned candidate links appropriately, and then assign the candidate links to FSOs in a greedy manner while minimizing the total area of the mirrors; here we use one overhead mirror (of appropriate size) to cover the reflection points in each cluster. We discuss each of these two steps in more detail below.

Clustering of Unassigned Candidate Links. Since an unassigned candidate link can be characterized by the corresponding pair of racks, the clustering of unassigned links can be looked upon as clustering of rack-pairs. Essentially, we wish to cluster the rack-pairs, such that two rack pairs (R_a, R_b) and (R_c, R_d) are in the same cluster iff any assigned candidate links connecting R_a to R_b or connecting R_a to R_b will have closely-located reflection points. The idea is based on the insight from Figure 6, wherein rack pairs (R_1, R_4) and (R_2, R_3) should be clustered together. Before describing the above clustering algorithm more formally, we define a term.

DEFINITION 2. (Reflection Polygon) For any pair of racks R_a and R_b, we define the reflection polygon as the convex hull of the reflection points of all possible (assigned) candidate links connecting an FSO on R_a to an FSO on R_b. See Figure 7. To construct the reflection polygon, we can just consider the sixteen reflection points corresponding to the candidate links connecting the FSOs on the "corners" of racks R_a and R_b, and then find the convex hull of these points. □

To cluster the rack-pairs, we first map each rack-pair to its reflection polygon. Then, we cluster the reflection polygons, and this in turn clusters the rack-pairs. We cluster the reflection polygons by "stabbing" them using a minimum number of points. Here, a point is said a *stab* a polygon if it lies inside the polygon. This is an instance of the well-known *stabbing problem* [22], and we use the standard greedy approach to stab the given polygon with a minimum number of points. The set of polygons that are stabbed by the same point are placed in the same cluster; if a polygon is stabbed by multiple points, then it is placed in one of the corresponding clusters randomly.

Assignment of Unassigned Candidate Links. Once the rack-pairs (and thus, the unassigned candidate links) have been clustered as above, we employ the following greedy approach to assign the candidate links to FSOs. Let the given set of unassigned links be E. In each iteration, we randomly pick a link (R_a, R_b) from E that has not been picked already, and assign it to the "best" pair of FSOs on racks R_a and R_b. The best pair of FSOs is the one that results in the minimum increase of the area of the overhead mirror currently being used to cover the reflection points of already-assigned links in the cluster of (R_a, R_b). If (R_a, R_b) is the first link being considered from its cluster, then the assignment to FSOs is done randomly. While doing the above, we also ensure that no FSO gets more than the maximum number of links allowed per FSO. Note that the above algorithm allows each link to be eventually assigned, thus, create a dynamic graph with maximum number of links possible.

After the above assignment, there may be more than r mirrors (due to more than r clusters). In such a case, to satisfy the problem constraint of r mirrors, we iteratively find and merge a pair of mirrors that results in minimize increase in the total area when merged together.

3.2 GM-Based DC Architectures

In this susbection, we address the problem in the context of GM-based FireFly architecture, where each FSO device is equipped with a GM rather than a number of SMs.

GMM Problem. The input to the problem is same as that to the SMM problem – except that each FSO is equipped with a GM. The goal and objective of the GMM problem is also same as the SMM problem except that the candidate links have another constraint, viz., there must be an orientation of the GM at each FSO a so that all the assigned links at a are covered by the GM.

Algorithm. The key difficulty in the GMMproblem is that for a candidate link (a, b) connecting a to b to be valid – the GM of $a$must "cover" b (i.e., the corresponding reflection point) and *vice versa*. Thus, it would be effect to orient the GMs in pairs or sets (as in the case of the GM-PCFT algorithm of [18] for the pre-configuration problem). In light of the above, we solve the GMM problem in the following three high-level steps.

1. As in the SMM problem, we consider a random simple graph over racks – here, since there is no limit on the *number* of links per FSO or rack, we consider a complete (simple) graph G over the racks, and try to assign as many links of G as possible (note that, in the GMM setting, we may not be able to assign all the links of G). As in the initial step of the SMM problem, we start with clustering the rack-pairs (or unassigned candidate links).

2. Second, we orient the GMs in a way that ensure the following: (i) there are an adequate number of FSO pairs that have their GM's covering *each other*, (ii) the candidate links in the same cluster (as determined in the first step) have closely located reflection points.

3. Third, we cover the reflection points of the *assigned* candidate links using a small number of overhead mirrors.

Since the first step is almost same as from the previous subsection, we describe in detail the second and third steps above.

Orienting GMs. As mentioned above, to ensure that we get an adequate number of candidate links, we orient GMs in sets. In fact, we can use the block-based heuristic from [18] for their GM-PCFT problem, with a slight variation. We briefly describe the block-based heuristic from [18] and then suggest our change. The *block-based heuristic* runs in m iterations, where m is the number of FSOs per rack. In each iteration, one FSO per rack is oriented as follows. First, the given set of racks are partitioned into disjoint *blocks*, such that each block of racks is co-located and small enough to be covered by a GM (when appropriately oriented) on any FSO in the DC. For a rectangular grid of racks, a simple grid-based partitioning scheme suffices to create these blocks. Next, we create a random block-level matching M, and for each edge (B_1, B_2) in the matching, we orient a GM on each rack of block B_1 (B_2) towards a B_2 (B_1). In the original block-based algorithm, the GM picked from each rack for orientation above was done randomly. However, in our case, we would pick a GM (not already picked) in a way that

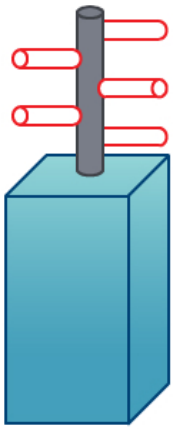

Figure 8: FSOs on a tower on rack

minimizes the sum of distances between the reflection points of the candidate links in the same cluster; this is the only change from the original block-based heuristic for our context.

Covering the Reflection Points of Assigned Links. At this point, we have the candidate links assigned to FSOs, and thus, all we need to do is to cover the reflection points on the ceiling with a small number of overhead mirrors. This can be looked upon as a set-cover problem, where the reflection points are the elements to cover and sets are the possible mirrors that can be placed on the ceiling to cover the points. We limit the number of possible mirrors to consider, by limiting the (i) shape of the mirrors to be rectangles, (ii) corner points of the rectangles to be one of the reflection points, and (iii) mirrors to be of an appropriate minimum size based on the density of the points to cover; we omit the details. Thus, the number of sets are now polynomial in the number of reflection points, and thus, we can employ a simple greedy approach to cover the reflection points using rectangles – we choose rectangles in the order of p/A, where p is the number of uncovered reflection points covered by the rectangle under consideration and A is the area of the rectangle.

4 LOS using Towers on Racks

One strategy to minimize obstructions without using any overhead mirrors is to use a small number (1-3) of "towers" at the top of each rack and place fixed steerable FSO devices at specific heights on these towers. See Fig. 8. This approach doesn't eliminate obstructions completely—since the FSOs and the towers may still obstruct with the beams. However, such obstructions can be reduced to a minimum by a careful (i) placement of towers on the rack, (ii) placement of FSOs on the towers, and (iii) assignment of links to FSOs. In the following subsections, we design techniques that maximize the number of links with clear line-of-sight (LOS) that can be assigned to the FSOs in the network. Also, to make the above LOS strategy based on towers viable, we need to design very sturdy towers that can withstand the weight of few tens of FSOs and preserve the alignment of the links. We believe that 2-3" diameter towers reinforced wtih carbon fiber rods can easily provide the required stability.

4.1 SM-based DC Architectures

In this section, we address the problem of establishing line-of-sight by placing FSOs on towers on racks, in FireFly architectures where the steering mechanism used is SMs.

SMT Problem. Consider a DC with n racks, each equipped with a certain number of towers, with each pole equipped with a certain number of FSOs. Each FSO is equipped with a certain number of SMs. The SMT problem is to (i) place, i.e., determine locations, towers on the top of the rack, (ii) place FSOs on each tower, and (iii) create a dynamic graph such that each candidate link has a clear/direct line-of-sight (note that there are no overhead mirrors for beam redirection). The optimization objective is to maximize the number of rack-pairs that have at least one candidate link between them in the created dynamic graph.

SMT Algorithm. As in §3.1 for the SMM problem, we start with a random simple graph over racks with mk links; however, since we may not be able to assign all the links due to occlusions, we try to maximize the number of links of G that can be assigned. Aa a post-processing step, we can add "duplicate" links between rack-pairs, if possible.

Our algorithm consists of three steps: (i) First, we place the towers on racks in a greedy manner, (ii) Then, we place FSOs on the towers at regular heights, and (iii) Finally, we prioritize links and assign them, in the order of their priority, to the FSOs in a greedy manner.

To determine the locations of the towers on the racks, we iteratively pick the best location on top of a rack to place a tower at. Here, a location is ranked based on the total number of already-placed towers that are visible from the location under consideration. During the course of this greedy approach, it is ensured that only the given number of towers are placed on any particular rack. Once the towers have been placed, we place given FSOs on the towers at *regular* heights.

Finally, we assign links to FSOs in a greedy manner – here, we prioritize the (unassigned) candidate links, and then consider them for assignment one by one in order of their priority. The priorities are assigned as follows. First, observe that the main reason an unassigned link may not be assignable is if the pairs of FSOs that are actually available, between the corresponding rack pairs, may not have a clear line-of-sight. Based on this observation, we prioritize the links in a way to avoid the above situation of not being able to assign a link. In particular, for each link (a, b), we compute its priority as the total number of pairs of FSOs that have a clear line of sight and can be used to assign the link (a, b). Since the weight may change over the course of the algorithm, we keep the weight of links updated.

Now, we consider the links for assignment to FSOs in the decreasing order of their weights. When assigning a link – we pick a pair of available FSOs that have a clear line of sight, and if no such pair exists, we skip the link and go to the next link in order of their weight.

4.2 GM-based DC Architectures

In this section, we address the problem of establishing line-of-sight by placing FSOs on towers on racks, in FireFly architectures where the steering mechanism used is SMs.

GMT Problem. The input to the problem is same as that to the SMT problem – except that each FSO is equipped with a GM. The goal and objective of the GMT problem is also same as that of the SMTproblem except that the candidate links have another constraint, viz., there must be an orientation of the GM at each FSO a so that all the assigned links at a are covered by the GM.

GMT Algorithm. First, we place the towers on the racks and FSOs on the towers as in the SMT problem. To orient GMs, we observe that obstructions due to towers and FSOs play a significant role in whether a candidate link can be assigned or not. Thus, here, we orient the GMs *independently* – rather than in pairs of sets as in the block-based heuristic described in §3.2. In particular, we prioritize the GMs, and then consider them for orientation one at a time in

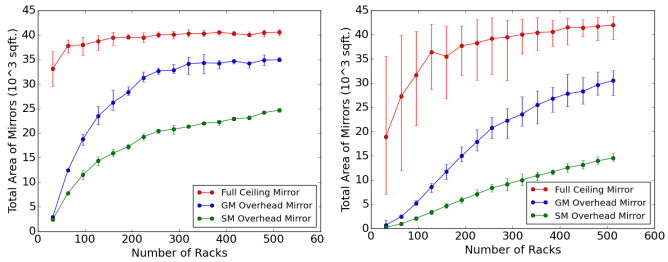

Figure 9: Total area of overhead mirrors in (a) random grid and (b) purely random layouts, for varying network size.

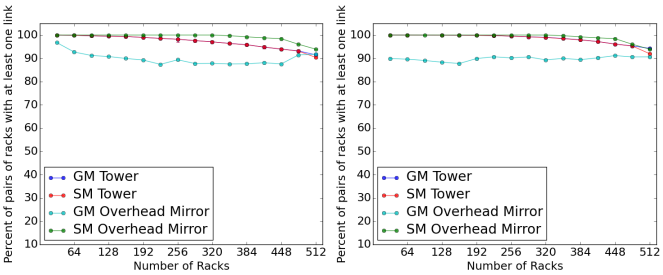

Figure 10: Percentage of rack-pairs with candidate links in (a) random grid, and (b) purely random layouts, for varying network size.

the order of their priority. For a network configuration that has sufficient number of FSOs on each rack, the above strategy should ensure that, for any rack, the set of GMs on that rack collectively "cover" most of the other racks; as this holds for every rack, in most cases, the above strategy should ensure that if a rack R_1 is covered by a GM g_2 on another rack R_2, then there is some GM g_1 on R_1 that covers g on R_2 – ensuring the desired candidate link between g_1 and g_2. This hypothesis is further confirmed by our simulation studies.

To assign priorities to GM, we count the number of racks that are "covered" by an orientation of GM – where a rack is considered to be covered by a GM, if there is an FSO on the rack that has a clear line of sign with the GM. We prioritize GMs based on its maximum count across all its possible orientations. When trying to orient a GM (when it turn comes), we pick the orientation that covers the most number of racks.

5 Evaluation

In this section, we evaluate our proposed techniques. We start with details of the simulations set-up.

Physical Parameters. The performance results of our proposed techniques depend on the physical layout of racks. Thus, we consider two different physical layouts: (i) *Random Grid of Blocks*, wherein "blocks" of racks are distributed randomly in the given area with the condition that they are axis-aligned; here, a block of racks consists of 8 racks arranged in a 2 by 4 grid, with a 5 feet gap between the two columns and an inch gap between the rows. (ii) *Purely Random*: In this layout, the racks are independently and randomly distributed in the given area; after locating each rack, we also rotate each rack by a random angle. We assume the top of each rack to be $2'$ by $4'$, which is sufficient to accommodate around 50 FSOs [18]. We assume the ceiling to be 10 feet above the top of the racks. We assume the total physical size of the data center to be five times the area taken up the racks; thus, the 512-rack DC is assumed to be in an area of 20,000 square feet.

For the overhead mirrors used in schemes of §3, we add a buffer of 1/2 an inch to all sides of the mirror, to accommodate minor link misalignments. For the tower-based approaches of §4, we assume towers of 2 inch diameter and four towers per rack. The total number of FSOs are divided equally among the four towers, and vertical distance between successive FSOs on a tower is 4 inches.

Network Parameters. By default, we run our simulations on a 512-rack network, with each rack consisting of 48 servers as in [18]. For the 512-rack, we equip each rack with 48 FSOs. For networks with smaller number of racks, we proportionally decrease the number of FSOs per rack. In all simulations, we assume the FSO device to be equipped with either 10 SMs or a GM with a coverage angle of 40 degrees. All communication links are robust 10 Gbps links. Finally, as in [18], we assume an overall reconfiguration latency of 20 msecs for the SMs as well as GMs steering mechanisms.

Traffic Models. As in [18], we use synthetic traffic models based on DC measurements [16,29]. As a baseline, we consider a `Uniform` model where flows between pairs of racks arrive independently with a Poisson arrival-rate λ/s, with an empirically-derived flow size distribution [16]. We use λ as the knob to tune the level of network saturation. Based on prior observations, we also consider the `Hotspot` model [16], where in addition to the `Uniform` baseline, a subset of rack pairs have a higher arrival rate λ_2 and a fixed large flow size of 128MB [29]. For `Uniform` loads, we use the label `Uniform X` where X is average load per server (in Gbps) by choosing suitable λ. For `Hotspot` loads, we use the label `Hotspot(Y,X)` where Y is the % of racks that are hotspots and X is the *additional* average load on each hotspot server; all `Hotspot` workloads use `Uniform 5` as the background traffic.

Total Area of Overhead Mirrors. We start with comparing the total area of overhead mirrors used by our techniques in §3 in comparison to the original FireFly architecture which uses a full-ceiling mirrror. We consider both layouts, viz., random grid of blocks and purely random, and plot the total area of mirrors used for varying number of racks. We allow up to eight times as many mirrors as the number of racks. See Figure 9(a)-(b). In both the plots, we observe the following: (i) The total area of mirrors used is significantly less then the full-ceiling mirror, especially for the SM-based architecture, (ii) The reduction for GM-based architectures is less than that for the SM-based architectures — this is expected, since the GM-based architectures have an additional constraint on the created links, (ii) Finally, the total area of the mirrors used is linear in the number of racks, which suggests the scalability of the schemes.

Percentage of Rack-Pairs with a Candidate Link. We now plot the percentage of rack-pairs that have a candidate link in the created dynamic graph, as it is an optimization objective of our schemes. See Figure 10(a)-(b). We make the following observations for both layouts: (i) As expected, the percentage is near 100% for the SM-mirror scheme, since we are able to assign all links of the "full" graph G, and the percentage of rack-pairs is slightly (about 10%) less for the GM-mirror scheme due to the additional link constraint, (ii) Surprisingly, for both the tower-based approaches, the percentage of rack pairs is near-optimal, i.e., very close to that of the SM-mirror scheme, which suggests the overall robustness of the tower-based approach, and (iii) With the increase in the network size, the percentage of rack-pairs with links decrease slightly – this is expected since occlusions are more in larger networks.

Network Performance Metrics: Average Throughput and Flow Completion Times. Finally, we evaluate the performance of our schemes in terms of network throughput and flow completion times for various traffic patterns. For this purpose, we implemented a custom flow-level network simulator that supports dynamic networks. For this simulation study, we use the 512-rack network, and plot the results for the grid-based physical layout (results were similar for

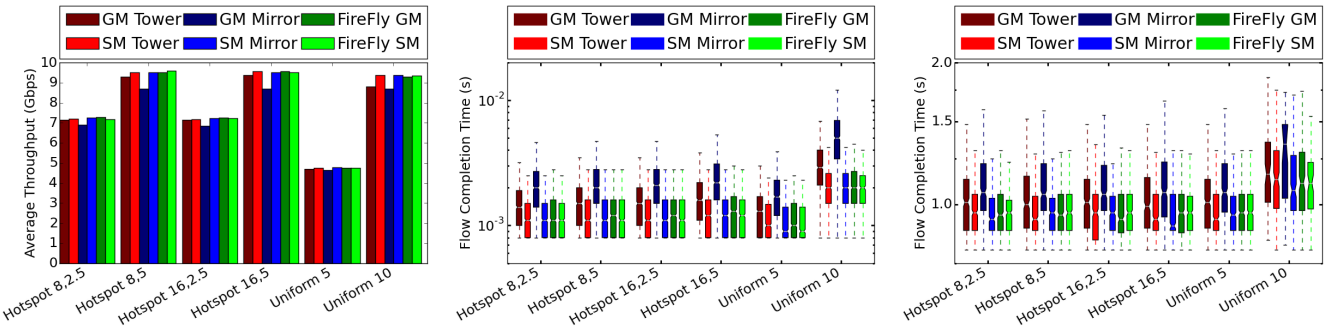

Figure 11: (a) Average throughput per server, (b) Flow completion times in short flows, and (c) Flow completion times in long flows, in a 512-rack DC for different schemes and traffic patterns.

the purely random layout). Figure 11(a) plots the average throughput per server; for the `Uniform` the average is over all servers whereas for `Hotspot` the average is over the hotspot servers. Figure 11(b)-(c) show the box-and-whiskers plots (showing maximum, minimum, median, 25%iles, and 75%iles) of the flow completion times (FCT) for short ($< 100MB$ flow size) and long flows respectively. The simulations results are based on a 30 sec run. We make the following observations: (i) the average throughput of all schemes is near-optimal (i.e., close to the FireFly's performance) for all traffic patterns, except for the case of GM overhead-mirror scheme which lags by about 5% in half of the traffic patterns, (ii) the distribution of the FCTs of our schemes is nearly identical to the FireFly architecture, except for the case of GM-based architectures which lag by 5-20% for the long flows and more for the short flows.

6 Conclusions

In this paper, we have proposed and evaluated alternative schemes to facilitate line-of-sight in FireFly, an FSO-based fully-wireless data center network. Our simulation results show the network performance of our proposed schemes is near-identical to the original FireFly's performance, which demonstrates the viability and attractiveness of the proposed schemes. FireFly has unprecedented benefits, but its use of a full-ceiling mirror limits it practicality. We believe that our proposed schemes and results remove one of the main obstacles to the feasibility of FireFly architecture in practice. In our future work, we plan to explore schemes that use non-flat mirrors in areas other than the ceilings and combine the use of use mirrors with our tower-based approach to yield more attractive schemes.

7 References

[1] A Simpler Data Center Fabric Emerges . `http://tinyurl.com/kaxpotw`.

[2] Cisco Global Cloud Index: Forecast and Methodology, 2012 to 2017. `http://tinyurl.com/7gnfeeb`.

[3] Data center survey. `http://www.flexiant.com/2013/09/04/report-shows-global-data-center-growth/`.

[4] Galvo mirrors. `http://www.thorlabs.us/NewGroupPage9.cfm?ObjectGroup_ID=3770`.

[5] Kent optronics, inc. `http://kentoptronics.com/switchable.html`.

[6] Magic quadrant for data center network infrastructure. `http://tinyurl.com/mpo3jzt`.

[7] NSA Utah Data Center. `http://nsa.gov1.info/utah-data-center/`.

[8] US government gives IBM cloud green light to serve agencies. `http://tinyurl.com/mx2v2mt`.

[9] D. Abts et al. Energy Proportional Datacenter Networks. In *Proc. ISCA*, 2010.

[10] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *ACM SIGCOMM*, 2008.

[11] L. A. Barroso and U. HÃűlzle. The case for energy-proportional computing. *IEEE Computer*, 40, 2007.

[12] K. Chen et al. OSA: An optical switching architecture for data center networks with unprecedented flexibility. In *NSDI*, 2012.

[13] C. Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, 32, 1953.

[14] A. Curtis, S. Keshav, and A. Lopez-Ortiz. LEGUP: Using heterogeneity to reduce the cost of data center network upgrades. In *CoNEXT*, 2010.

[15] N. Farrington. Optics in data center network architecture. `http://nathanfarrington.com/papers/dissertation.pdf`.

[16] A. Greenberg et al. VL2: A scalable and flexible data center network. In *ACM SIGCOMM*, 2009.

[17] D. Halperin et al. Augmenting data center networks with multi-gigabit wireless links. In *ACM SIGCOMM*, 2011.

[18] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. Das, J. Longtin, H. Shah, and A. Tanwer. FireFly: A reconfigurable wireless data center fabric using free-space optics. In *ACM SIGCOMM*, 2014.

[19] J. Hamilton. Datacenter Networks are in my Way. `http://perspectives.mvdirona.com/2010/10/31/DatacenterNetworksAreInMyWay.aspx`.

[20] B. Heller et al. ElasticTree: Saving energy in data center networks. In *NSDI*, 2010.

[21] D. Kedar and S. Arnon. Urban optical wireless communication networks: The main challenges and possible solutions. *IEEE Communications Magazine*, 2004.

[22] S. Kovaleva and F. C. R. Spieksma. Approximation algorithms for rectangle stabbing and interval stabbing problems. *SIAM Journal on Disc. Maths*, 20(3), 2006.

[23] J. Mudigonda, P. Yalagandula, and J. C. Mogul. Taming the flying cable monster: A topology design and optimization framework for data-center networks. In *USENIX ATC*, 2011.

[24] R. N. Mysore et al. PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric. In *Proc. ACM SIGCOMM*, 2009.

[25] L. Popa et al. A cost comparison of datacenter network architectures. In *CoNEXT*, 2010.

[26] A. Singla et al. Proteus: a Topology Malleable Data Center Network. In *HotNets*, 2010.

[27] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *NSDI*, 2012.

[28] G. Wang et al. c-Through: Part-time optics in data centers. In *ACM SIGCOMM*, 2010.

[29] X. Zhou et al. Mirror mirror on the ceiling: Flexible wireless links for data centers. In *ACM SIGCOMM*, 2012.