1 Introduction

Data centers (DCs) are a critical piece of today's computing infrastructure that drive key networked applications in both the private [] as well as the public sector []. The key factors that have driven this trend toward massive large-scale consolidation of computing power is the economies of scale that these offer, reduced or amortized management costs, better utilization of hardware via statistical multiplexing, and the ability to dynamically and elastically scale applications in response to changing workload patterns.

An efficient and robust *datacenter network fabric* is a fundamenal requirement that underlies the success of such large-scale datacenters. In particular, the design must ensure that network performance does not become a bottleneck for high-performance and high-availability applications [**?**]. In this context, data center network designs must satisfy several potentially conflicting goals: high performance (e.g., high throughput and low latency) [2, 4]; low equipment and management cost [2, 14]; robustness to extremely dynamic traffic patterns [5, 6, 8, 15]; incremental expandability to add new servers or racks [3, 16]; and a host of other practical concerns including cabling complexity, power, and cooling [9, 11, 13].

Meeting these requirements is as critical to the computing ecosystem as it is challenging. Existing DC architectures can be categorized into three classes: (1) *overprovisioned* fabrics (e.g., fat-trees or multi-stage Clos networks) that provide full-bisection bandwidth []; (2) *oversubscribed* fabrics (e.g., traditional "leaf-spine" where links higher in the hierarchy are oversubscribed) []; and (3) *augmented* fabrics where an oversubscribed core is augmented with reconfigurable wireless [5,8] or optical links [**?**]. The first two classes offer undesirable points in the cost-performance space—overprovisioning incurs high cost and concerns with respect to incremental expandability, while oversubscription can lead to poor performance especially in the "tail" [**?**]. While augmented fabrics promise a middleground, the specific approaches they adopt limit the degree of flexibility. Furthermore, all of these architectures and other extensions [**?**, 16] suffer from a fundamental problem of *high cabling cost and complexity* [**?**]. (We elaborate on this in Section **??**.)

Our vision: Rather than try to incrementally improve the poor cost-performance tradeoffs and high cabling complexity of existing architectures, we propose an *extreme* design point—a *flexible all-wireless inter-rack fabric*. We eliminate the overprovisioned/oversubscribed "wired core" altogether. Our intuition here is that topological flexibility (if done right) can replace the need for overprovisioning.

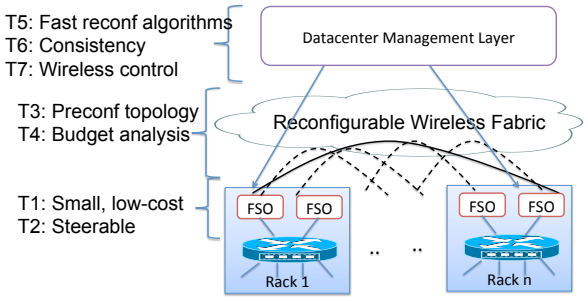


Figure 1: Overview of the Firefly vision

This vision would provide unprecedented degrees of flexibility. It will allow operators to dynamically reconfigure the *entire* topology to adapt to changing demands, as opposed to a few links []. Moreover, am all-wireless architecture eliminates maintenance and operational overheads (e.g., obstructed cooling) due to cabling complexity [**?**]. Furthermore, this can facilitate topology structures that would otherwise remain "paper designs" due to the perceived cabling complexity. Finally, a flexible architecture offers other quantitative benefits with respect to energy and cooling costs [] and incremental expandability [].

To realize this goal, however, we need to look beyond traditional radio-frequency (RF) based (e.g., 60GHz) wireless solutions as they are fundamentally limited in their range, capacity, and interference characteristics. In particular, we rely on *Free-Space Optical communications* (FSO) as it offers very high data rates (tens of Gbps) at long ranges, with low transmission power, and with very low interference footprint. Figure 1 shows a conceptual overview of our vision called Firefly.[1] Each top-of-rack switch is provisioned with some flexible FSO links that can be pre-configured to reach a subset of the other racks. A datacenter

[1] Firefly stands for .

management layer reconfigures the network topology in near real-time to optimally adapt to the current traffic workloads.

Research plan and Intellectual Merit: We need to address fundamental algorithmic, networking, and system design challenges along three main thrusts (Figure 1) in order to turn it into reality.

- Datacenter scale deployments impose new form-factor, cost, and steerability requirements for FSOs that are fundamentally different from the traditional long-haul use-cases. Thus, we need to developing cost-effective commoditizable solutions (**??**) that can be steerable at very fine-grained timescales(**??**).
- The use of a flexible topology needs new algorithmic foundations for reasoning about flexible network design (**??**). Furthermore, steerable FSOs raise unique economic, physical, and geometric constraints (**??**).
- Our vision raises unique network management challenges and requires novel algorithms for joint topology and traffic engineering (**Task 3**, mandates new consistency abstractions that guarantee reachability and performance when links may be in flux (**Task 4**), and requires new wireless mechanisms to replace traditional wired control channels (**Task 5**).

Team Qualifications The proposed research spans aspects of optical technologies, electrical and mechanical steerable technologies, algorithmic foundations of networking, wireless networks, and network management. Our team comprises three computer scientists and one mechanical engineer with complementary expertise spanning the domains of wireless networking [], network management [], software-defined networking [], and the use of laser-based optical technologies []. Our proposed research is highly integrative and the expertise of the PIs complement each other in addressing the algorithmic and system-design challenges involved in Firefly. The PIs have a strong history of collaboration that has resulted in multiple publications and outreach activities [].

2 Motivation and Research Overview

As our title suggests, there are three key aspects to our vision: *flexibility*, *wireless*, and the use of *free-space optics*. We begin by arguing the need for each of these aspects before discussing a high-level view of our proposed architecture.

2.1 Case for Flexibility in Datacenters

The critical role that datacenter performance plays has motivated several efforts in the research literature. At a high-level we can summarize these efforts along two dimensions: (1) the extent of oversubscription in the "core" and (2) the type (if any) of flexibility it offers to reconfigure the inter-rack topology based on the offered load. Table **??** presents a high-level taxonomy of prior proposals for this inter-rack *fabric* along these two key dimensions.

In this taxonomy, traditional datacenter fabrics such as leaf-spine or fat-tree architectures offer no flexibility construct a static topology. This inflexibility offers extreme points in the space of cost-performance tradeoffs. Furthermore, the overprovisioned networks typically rely on structured graphs that may additionally impact the ability to

Category	Backbone	Flexibility	Notes
Leaf-Spine (e.g., [])	Wired, oversubscribed	None	Poor performance
Full bisection bandwidth (e.g., [2,4])	Wired, no oversubscription	None	High cost + cabling complexity, no incremental expandability
Wireless augmentation (e.g., [**?**,8])	Wired, oversubscribed	Few 60Ghz links	Low range, bandwidth
Optical augmentation (e.g., [**?**,**?**])	Wired, oversubscribed	Single optical	Limited flexibility, Single point of failure
Firefly vision	None	Steerable FSO	Not commodity yet

Table 1: Taxonomy of datacenter network architectures and recent research proposals

incrementally expand the datacenter [16]. More recent work analyzed real datacenter workloads and show that the traffic patterns are quite variable and exhibit hotspots of inter-rack activity (e.g., one-to-one, one-to-many), and tha a few "heavy" flows contribute a significant share of the total traffic [**?**,4,5]. These led to *augmented* architectures where an oversubscribed core is extended with a small number of reconfigurable optical connections [] or wireless links [] to accommodate these hotspot demands.

While our vision builds on these data-driven insights from prior work, we argue that the aforementioned efforts simply don't go far enough. Rather than use flexible links to incrementally improve the performance of an existing oversubscribed network, we posit that extreme flexibility, if done suitably, can obviate the need for overprovisioning and the need for a static backbone! Furthermore, this flexibility can enable new dimensions of cost and energy savings by selectively shutting down links/switches depending on the network load [**?**, **?**].

To provide the basis for this intuition, we consider an abstract model of a *flexible* datacenter as follows. We consider a data center of 20 racks, where each rack has l machines. We use 1Gbps $2 \times l$-port switches, as in FatTree architectures. The ToR (top of rack) switches use l ports for the machines, and the remaining l ports for inter-switch connections. The non-ToR switches use all their ports for inter-switch connections. Our fixed architecure for (a) is based on a random graph (of inter-switch connections) over XX number of switches, and delivers a performance of ZZZ flow-completion time. We generate D-flexible architecture as follows: We allow D ports of each ToR switch and $2 \times D$ ports of each non-ToR switch to be "reconnected" at each epoch; the interconnections between remaining ports are random but *fixed*. Thus, higher the value of D, more flexible is the architecture. We also consider P-ToRFlexible architectures, wherein we use only ToR switches with P ports each; l ports of these ToR switches are connected to the rack machines, and the remaining $P - l$ ports are reconnected at every epoch.

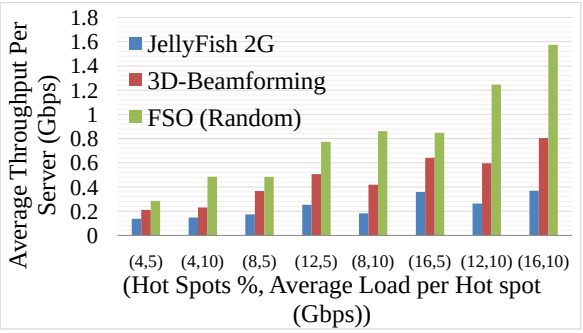


Figure 2: The case for flexibility – it can provide performance comparable to a full bisection bandwidth network with a lot less equipment and even get rid of aggregation layers. PLACEHOLDER from hotnets

Figure 2 shows that by increasing the degree of flexibily .

2.2 Case for Wireless via Free-Space Optical Communications

Why wireless? In order to realize such a flexible fabric, ideally we would like a massive reconfigurable "patch-panel" that can interconnect pairs of racks on-demand [**?**]. Of course, such a big-circuit-switch abstraction is infeasible on two fronts. First, this would need to have very high degree and internal backplane needs to be petabit-scale; e.g., with $n = 500$, $D = 20$ and 10 Gbps links this backplane needs to run at $(D \times n)^2 \times 10\ Gbps = 10^{10}\ Gbps$ []. Second, the cabling complexity of this network would be prohibitively high []. As argued elsewhere, cabling complexity creates other operational overheads in terms of failure and interferes with cooling and airflow considerations []. For the same reasons, we believe all-optical designs while conceptually appealing, are not viable. Furthermore, this giant optical switch introduces a single-point of failure [**?**, **?**, 15].

To avoid the need for such a massive patch-panel and the need for large pure optical switches, we turn to *wireless* links between the ToR switches to create a flexible inter-rack fabric. Note that we are not proposing a fully wireless data center as in prior work [7]; our focus is on the "inter-rack' fabric. Thus, we envision

each ToR switch will be equipped with D wireless transceivers as shown in Figure 1.

Why FSO? Unfortunately, traditional radio-frequency (RF) wireless technologies (e.g., 60GHz) suffer many shortcomings: RF links produce a large interference footprint due to a wide beamwidth, even with new "ceiling mirror" architectures [8]. Moreover, beam-steering (needed to impart flexibility) technologies for RF links are either slow and inaccurate [8] or generate even larger interference footprint [**?**]. Last, the data rates of RF links fall off rapidly with distance [8]; higher transmit power can increase the range, but also yields higher interference and energy usage, and is ultimately limited by regulations.

To overcome the limitations of RF technology, we turn to a somewhat non-standard "wireless" technology, namely the use of free-space optics (FSO). FSO communication [10] uses modulated visible or infrared (IR) laser beams in the free space to implement a communication link. Unlike traditional optical networks, the laser beam in FSO is not enclosed in a glass fiber, but transmitted through the air. There are two main benefits of FSO compared to traditional RF technologies that make it a promising candidate for data centers:

- *Low Noise and Interference.* Unlike RF links, FSO links do not suffer from multipath fading or EM noises []. Moreover, FSO's beam divergence is many orders of magnitude narrower than RF (in the order of milliradians or less (1 milliradian = $0.0573°$). This reduces 'interference footprint' to a negligible level. Thus, FSO communications from multiple senders do not interfere, unless they are aligned to the same receiver (which is easy to avoid).
- *High Data Rates over Long Ranges:* Optical communications inherently provide significantly higher data rates than any existing RF technology owing to the use of much higher frequency and absence of regulatory restrictions [10]. Coupled with much lower attenuation of power over distance, FSO links are able to offer data rates in the Gbps to Tbps regime at long distances (several kms) even with modest transmit power (watts) [10]. Commercially available FSO devices already offer data rates of 2.5 Gbps [1], and demonstration systems even report data rates in the order of Tbps [12] – both for long distance (in the order of km) links.

2.3 Proposed Research

Figure 3 summarizes the above discussion on how the three key aspects of the Firefly benefit different considerations of datacenter network design. In summary, flexibility is key to ensuring high-performance while minimizing equipment cost and also acting as an enabler for energy reduction []. An all-wireless architecture eliminates concerns about cabling complexity and its interference with cooling solutions [**?**]. Finally, by leveraging FSO technology, we effectively eliminate concerns w.r.t range, interference that can impact the performance of a wireless interconnect. That said, we have to address three fundamental challenges in order to turn this qualitative appeal into practical quantifiable benefits:

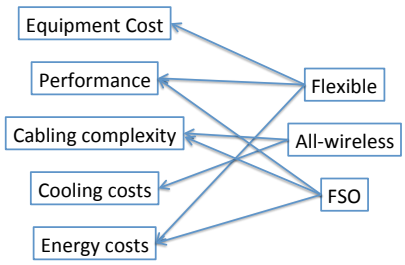


Figure 3: Overview of the key concepts underlying Firefly and how they benefit different aspects of datacenter considerations from Section 1

Feasibility of Precise and Fast Steering (Section 3): We must first demonstrate the feasibility of a small, cost-effective, and energy-efficient FSO device that is capable of delivering 10-100Gbps data rate. Second, we need to design and/or investigate mechanisms that will steer the laser beam with extreme precision (within microradians) and high speed (order of few tens of milliseconds).

Algorithmic foundations of flexible topology design (Section 4): Due to the limited steerability of the considered steering mechanisms, we need to *pre-configure* the steering mechanism of each FSO device (by pre-aligning or pre-orienting it) to a set/range of other FSO devices. This pre-configuration determines the set of fixed candidate links, from which the active links are chosen in real-time (called *reconfiguration*) depending on the prevailing traffic.

Effective datacenter management (Section 5): Finally, we need to address practical system design challenges in realizing a practical datacenter-scale management layer. To this end, we leverage recent advances in software-defined networking to achieve fine-grained control over the network configuration. However, we need to ensure that the reconfiguration latencies are very low and also engineer mechanisms that guarantee the network operates consistently in spite of the topology change-induced flux. Furthermore, our all-wireless vision eliminates conventional assumptions about reliable out-of-band control channels, and thus we need new wireless control channels as well.

3 Designing FSO Links for Flexible Inter-Rack Networking

Our goal in this section is to present the design of FSO transceiver modules that have the needed proper-ties for the design of a flexible inter-rack fabric, i.e., (i) size, power and cost effectiveness, (ii) ability to provide 10-100Gbps data rate, (iii) ability to align with the needed precision, (iv) ability to steer precisely and with low latency. The conventional use case for FSO has been terrestrial long distance (miles) or even satellite and space communications (1000s of miles or more) []. Commercially available FSO systems for these applications do not optimize on size, power or cost.[2] The reason for this is that they have to over-come several outdoor challenges in laser propagation (e.g., beam path variations due to scattering from fog or dust and due to significant temperature/humidity variations along long paths, larger transmit power requirements to account for path loss as well as divergences over long distances, non-trivial alignment problems due to structural swaying etc). While these challenges largely go away in datacenters, several new requirements arise. The links will now need to be able to steer with very low latency for topology reconfiguration. Size, cost and power now become central issues: Firefly will have thousands of FSO links and real estate on the top of the rack for deploying the transceivers are limited. Further, the design must be cost effective and power efficient — an argument put forward in Section 2.

Due to these unique needs, we have proposed a ground up design. However, we need to still rely on commodity components as much as possible to make an entire end-to-end prototype feasible within the budget/timeline of the project (described in Section 7). While this prototype will be of small scale, the description below articulates a pathway for designing scalable systems.

3.1 Design for Low Cost, Size, Power and Reliability

An FSO communication link consists of two basic components: i) a transmitter (TX) that is a modulated laser source (typically infrared), ii) a receiver (RX) that is a high-speed optical detector along with a de-modulator, iii) the optical path. A major step in our design is that we propose to repurpose optical small form-factor pluggable (SFP) transceivers [] for TX and RX components. Optical SFPs are used to interface optical fibers with (electrical) packet switches. The advantage of this approach is that we can exploit the large commodity market of optical SFPs that already operate at 10+ Gbps. This keeps the cost low in relation to a first-principles design of TX and RX. The optical SFPs are also very small in size (state size??) and already contain very reliable, field-tested TX and RX components. They are also indeed low power (state power??). In any case, datacenters often use them regardless of the network architecture, whenever optical fibers need to be used to support high data rates at distances that standard copper cables are unable to carry []. Thus, there is no added power burden from our use of optical SFPs.

Having just TX and RX are not enough. The optical SFP is designed to interface directly to an optical fiber[3] with the fiber confining the beam via a series of internal reflections. Instead now the laser beam will launch directly into free space requiring us to design the optical path.

[2] Typical commercially available systems for terrestrial applications [**?**] are roughly 2 cubic feet, costs $5-10K for a single link and consumes XXX watts.

[3] Typically, a fiber pair for bidirectional communications. Single fiber SFPs [] are also possible where the laser and detector at each end point are interfaced to the same fiber with two directions operating at two different wavelengths. As would be apparent later, use of single fiber SFPs make our approach easier as only one optical path needs to be designed per FSO transceiver.

Optical Path Design From any laser source the beam diverges in a cone as it propagates quickly losing power. This divergence is typically arrested using a suitably designed collimating lens on the optical path near the TX. This makes the laser beams roughly parallel, now diverging at a very slow rate in the order of milli-radians or less. Finally a similar lens near the RX focuses the beam back onto the detector. From basic optics, an inverse relationship exists between the diameter of the propagating laser beam at the so called beam waist (the narrowest part of the beam near) and the rate at which it diverges beyond this point (divergence angle). Thus the optical design is critical so that a beam of the right diameter at the waist can be formed. While larger beam waist are better to keep the divergence negligible this also requires a larger lens (size issue); smaller diameters on the other hand may make the beam diverge too quickly for it to be strong enough at the extreme distances within the datacenter (say, 100m). This tradeoff is crucial in designing the FSO links in Firefly and influences the optical design. Beam diameter also influences alignment challenges that we will describe now.

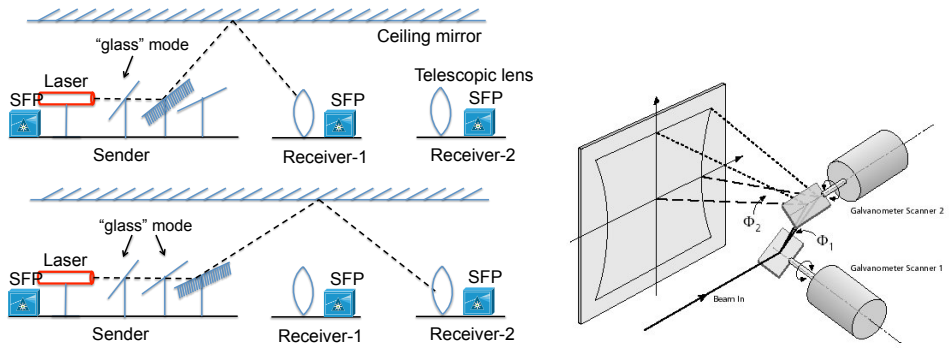
Alignment The optical detector at RX gathers energy proportional to its size (in the order of XX for SFPs). Ideally, we do not want the beam to have a much larger diameter at the RX than the detector size lest only a very small fraction of the power will be captured that may not be enough for reliable detection. Thus, both large diameter beam waist and large divergences are poor design choices. On the other hand, large diameter helps in alignment as small shifts in the optical path will have negligible effect in the detector so long as the total energy received at the detector is above the detection threshold. This helps recovering from rack vibrations, jarring effects due to maintenance, optical path drifts due to air temperature varia-tions, and other similar issues. If the beam diameter itself is not sufficiently large to account for the above, small positioning corrections of the TX and/or RX will be needed. We plan to use piezoelectric positioners or thermal heaters with a high-thermal-expansion material can provide these needed precision corrections at reasonable costs. The feedback needed to these corrections can be obtained from the DOM (digital optical monitoring) support already present in the optical SFP standard and carried on the I2C bus via the connectors on the module. If corrections on the RX side are insufficient and the TX side needs to be adjusted as well, the RF-based control mechanism will be used (discussed in Section 7).

In a datacenter, the space above the racks is a natural choice for laser propagation as this space is roughly free from obstruction. The FSO devices will be anchored on the top of the rack and connected to ToR switch ports. The devices themselves will either be staggered so that they themselves do not present ob-structions or mirrors in the ceiling (and possibly additional mirrors on the beam path for redirection) will be used to avoid obstructions. Overall, we anticipate that a single FSO assembly including the alignment and beam redirection machinery can be put together within about 3"x8" footprint such as that a few tens of such devices can be packed on the top of a rack. A critical task in the project will be designing the op-tical elements (e.g., mirrors, lenses) and their precise positioning such that they operate for wide range of distances, e.g., few meters for neighboring racks or over 100m for distant racks in a large data center.

3.2 Re-configurability via Beam Path Redirection

In Firefly the ToR switches are to be interconnected via the FSO links to create an FSO-based inter-rack fabric. To achieve reconfigurability in this fabric, the beam from the TX must fundamentally be able to redirect between multiple possible RXs on top of other racks. We have investigated two mechanisms to achieve this, i) a dedicated alignment approach via switchable mirrors and ii) a beam steering approach via Galvo mirrors. They both present certain tradeoffs. The impact of such tradeoffs in the overall performance is unclear without a careful experimentation. Such an evaluation will be part of our work. We anticipate that we will pick one or a combination design in our final prototype and in future research.

Approach I: Dedicated-Alignment – Switchable Mirrors In a dedicated alignment approach each beam path is manually oriented statically and remains fixed dur-ing normal operation. The active beam path is then selected among several candidate beam paths. One approach to do this is to use switchable mirrors (SMs)



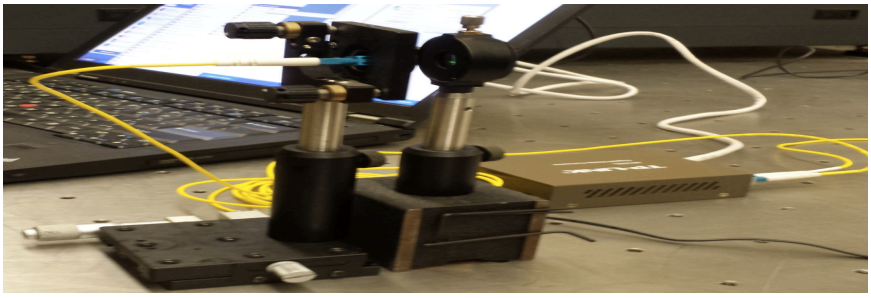
(a) Switchable mirror [sd] update fig. (b) Galvo mirror

Figure 4: Beam re-direction approaches. (a) A dedicated-alignment beam re-direction approach using SM. (Top) The second SM is in reflection mode, redirecting the beam to Receiver 1. (Bottom) The second mirror is now transparent and the third mirror is in reflection mode, directing the beam to Receiver 2. (b) Beam steering with GM. Two mirrors direct incident beam into a rectangular cone.

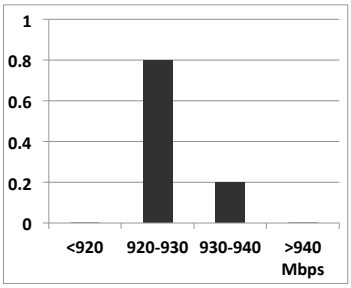
made from a special liquid crystal material that can be electrically controlled to rapidly switch between reflection (mirror) and transparent (glass) states [**?**]. Referring to Figure 4(a), each FSO device will be equipped with multiple SMs. Each SM will be pre-aligned to a dedicated beam path. The desired link is established by placing one of the SMs on the TX in the mirror state and the other SMs in the transparent state. An analogous arrangement will be made at the other end (not shown). At any instant, only a subset of candidate links is active (one per FSO) based on the SMs states. A mirror fastened to the ceiling redirects the beam back to the receiving rack, making efficient use of the above-rack space while minimizing interference. The pre-alignment for each SM is done statically at configuration time. This is directed by pre-configuration topology design (Section 4). A custom, hand-held mechanical adjusting mechanism can be used that will be attached temporarily to SM assembly and adjust each mirror along the calculated orientation.

Approach II: Beam Steering – Galvo Mirrors In beam steering, the optical system can dynamically re-direct the beam on command and in real time. This requires computer-controlled movable optics. Among the many possible candidates, one promising approach is to use a galvo mirror (GM) system. Shown in Figure 4(b), two computer-controlled, motorized mirrors are mounted at right angles direct incident beam into a rectangular cone. The (fixed) incident beam can thus be directed into a rectangular cone under computer control. Commercially available systems [] can provide a cone half angle (Φ_1 and Φ_2) of $\pm 20^\circ$, for a total rectangular cone angle of 40 in both directions. A typical pointing accuracy is within 15 μrad [], resulting in a beam positioning precision within 1.5mm for beam paths of up to 100m. GM provides some advantages relative to SM. Since GM can steer on a continuous scale no additional alignment mechanism is necessary as in SM. Also, GM can re-direct the beam to any number of FSOs within the cone in contrast to SMs that provide only a small number of beam paths. The re-direction latencies are however comparable or better in GMs. However, the limitation of GMs is a limited steering angle that makes the network topology dependent on layout geometry and rack locations. To address this pre-alignment can be used just like SMs, or a third servo and mirror combination can be used to increase the angle. Commercially available GMs are also more expensive than SM, with cost dependent on precision and speed. Commercially available GMs not have the form factor we desire (as they have a different use case), but a custom design is certainly possible to achieve custom angles and form factors. [**?**]

Integrating Emerging Technologies The previous design approaches are based on technology available today, to provide a guaranteed path forward. During the project, however, an exhaustive search of emerging and less-known technologies will be conducted to explore additional options in the photonics and optoelec-



(a) Experimental prototype [sd] placeholder fig.



(b) Link characterization

Figure 5: (a) Experimental prototype showing FSO communication using SFP. (b) Distribution of per-second TCP throughputs (in Mbps) over a continuous 30 hour period on the FSO link (1Gbps) over 7.5m.

tronics communities for beam steering, alignment, and range. Examples include MEMS-based switchable mirrors [], low-cost, high range-of-motion servos, low-cost, highperformance aspheric focusing optics (as used in phone cameras), and custom-designed systems built in-house.

3.3 Preliminary Work: Proof-of-Concept Prototyping

A proof-of-concept prototype has been developed to demonstrate free space operation using SFPs. See Figure 5. It uses a pair of 1Gbps SFPs using 1310nm laser. Instead of launching the beam directly from the SFP we launch from a single mode optical fiber that is connected to the TX SFP on one end with the other end terminating in free space. Due to the narrow $8 - 10\mu$m fiber diameter the initial beam divergence is very large. An achromatic doublet lens is used to collimate the beam to a roughly 4mm diameter waist with the fiber tip positioned at the focal point of the lens. An optical bench and translating mounts help in the positioning. The collimated beam propagates up to a distance of 7.5m where an identical lens re-focuses beam on the detector of the RX SFP. Since the SFP used here uses two separate optical paths (for duplex operation), the return link is closed using a regular fiber. This way standard network protocols can be run to characterize the free space link. Two laptops are connected to the SFPs via media converters and TCP throughput experiments (with forward direction going over free space) are run for over 30 hours continuously. See the distribution of per sec throughputs in Figure 5 that demonstrates an extremely stable link. It does not show any statistical difference from the wired case (both links using fiber). The quality of link is also studied when the TX-RX are misaligned. The TCP throughput is stable up to a transverse shift of ± 0.7mm showing a great promise in addressing alignment issues. Beyond this the throughput drops sharply going to zero within another 0.1mm.

We have also built a proof-of-concept prototype to evaluate the viability of switchable mirrors. Here, we have used a 12" x 15" switchable mirror (SM) from Kentoptronics [**?**] tuned for the IR spectrum; and normal mirrors. The switching latency of the SM is found to be around 250 msec. Because the switching latency is proportional to the SMs surface area [**?**], we estimate a < 5 msec latency for a small (1" x 1") SM we propose to use. Finally, we have confirmed that the FSO beam can be reflected from conventional mirrors with no loss in TCP throughputs even after multiple reflections.

4 Pre-Configured Flexible Topology Design

Our overarching goal is to design the most cost-effective and efficient flexible network design that can work within these constraints. In this respect, the hardware elements of our Firefly architecture, discussed in the previous section, impose physical and geometric constraints on the network design. For instance, the size of the FSO device assembly limits the number of FSOs that can be placed on the rack and the cost/range of steering mechanisms may also come into play.

Irrespective of the steering mechanism we will use, there are fundamentally two different timescales of operation that the hardware design imposes. First, at a *coarse* timescale, we need to *pre-configure* the steering solutions to define the space of *potentially realizable* topologies. For instance, this entails choosing the specific alignments of the different SMs or scoping the beam angle of the GMs. These operations will likely involve some offline tuning possibly with some huma involvement; e.g., to operate the pre-configuration machinery. Second, given this pre-configuration step, we can do a more fine-grained selection (i.e., at the timescales of a few millseconds) to choose a *specific real-time configuration.* Corresponding to these timescales, we envision the overall network design workflow shown in Figure 6. At a coarse scale (say on the scale of a few days or planned events), we reconfigure the Firefly fabric that defines the space of *pre-configured topologies* (PCT) and then we run a real-time management system that chooses one among the realizable topologies (see Section 5).

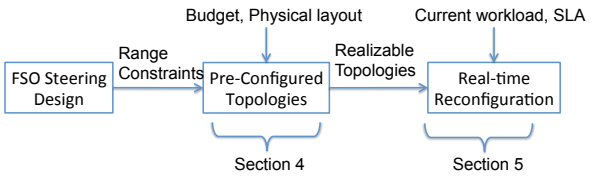


Figure 6: Interaction between the design constraints induced by FSO steering choices, the selection of preconfigured topologies, and the real-time topology selection

In this section, we focus on the coarse timescale selection problem to choose the set of pre-configured topologies that can provide the optimal performance, given the physical constrants, pricing of hardware devices, and overall budget. At a high-level, this means that we need to determine a range of network parameters—number of machines per rack, number of FSOs per rack, number of FSOs equipped with GM vs. SM, number of SMs per FSO, the pre-orientation of GMs, and the pre-alignment of SMs. The pre-orientation of GMs and pre-alignment of SMs in the system define the set of *candidate links* (between pairs of FSOs). This high-level problem is significantly different from prior theoretical work in network design [] on two fronts: (1) the *flexibility* to rewire the network introduces new dimensions that requires us to rethink traditional metrics of "goodness" as well as corresponding algorithms for topology design; and (2) the unique physical, budget, and geometric constraints of the Firefly deployment. Thus, we break down the above problem into two steps. First, to gain insights into the combinatorial and theoretical nature of the problem, we focus on a more general problem of designing an optimal "pre-configured flexible topology" (*PCFT*) given the number of racks, number of FSOs per rack, and maximum number of candidate links per FSO (without reference to GMs or SMs), and then in Section 4.2 we use the algorithm(s) for this problem to solve the budget-based network architecture design optimization problem.

4.1 Foundations of Pre-Configured Flexible Topology Design (*PCFT*)

To gain insights into the theoretical foundations of the problem, we abstract away the details of the SM or GM or the cost budget and focus on the following problem: Given the number of racks n, number of FSOs m on each rack, and the maximum number of *candidate links* k per FSO, the *pre-configurated flexible topology* (*PCFT*) design problem is to determine the set of candidate links for each FSO, so as to optimize the "dynamic bisection bandwidth" of the network (defined below).

Task 1: *We will investigate the theoretical foundations of flexible topology design that maximizes dynamic bisection bandwidth in conjunction with other measures of network goodness (e.g., diameter).*

Rethinking Metrics for Flexible Topologies: The traditional bisection bandwidth metric [] reflects a *static* perspective of the topology. However, in our context, we should instead consider the notion of *dynamic* bisection bandwidth (DBW) since different realizable topologies can be used for different communication requirements. Formally, the dynamic bisection bandwidth of a given pre-configured topology Π can be defined as follows. Let T be the set of realizable topologies of a given pre-configured topology Π, P be the set of partitions of the given network into two equi-sized sets of machines, and $BW(t, p)$ be the bandwidth of the topology t for the partition p. Then, the *dynamic bisection bandwidth* for the pre-configured topology

Π, denoted by DBW(Π), is defined as:

$$\text{DBW}(\Pi) = \min_{p \in P} \max_{t \in T} \text{BW}(T, p).$$

In addition to bisection bandwidth, a bound on worst-case latency between pairs of network nodes has also been considered as an objective in designing datacenter topologies [**?**]. Worst-case latency can be approximated by the network diameter. In our context, we can define an appropriate notion of *dynamic diameter* (as for DBW above), and consider maximizing DBW under the constraint of bounded (dynamic) diameter.

Finally, if we have some coarse statistics available on the expected inter-rack traffic, then we can use it to tailor our DBW objective accordingly. Note that since pre-configuration can only be done on an infrequent basis (e.g., weekly), so we are only interested in coarse traffic knowledge; the near-term traffic information is instead used for reconfiguring the network (discussed in Section **??**). One simple form of inter-rack traffic statisfics could be in the form of a weights between every pair the racks, and the DBW definition can be appropriately tailored as in [**?**] for multi-commodity min-cut. In our research, we will also consider more sophisticated stochastic traffic models [].

Connection to Known Graph Problems: In general, the PCFT problem is in the class of the *network design problem (NDP)* [**?**], wherein the goal is to extract a subgraph satisfying some design criteria and optimizing a given objective function. More specifically, the PCFT problems falls in the class of *degree-constrained subgraph* problems [**?**] wherein the extracted subgraph is constrained by the degree on each vertex. What distinguishes the PCFT problem from the prior-addressed NDP problems is the choice of our objective function, viz., dynamic bisection bandwidth.

Note that the PCFT problem under DBW maximization is very different from the well-known NP-hard problem of *computing* the bisection bandwidth of a *given* graph []. The special case of the PCFT problem, when $k = 1$, actually boils down to constructing an m-regular graph over n nodes with maximum (static) bisection bandwidth. The closely known problems to this $k = 1$ case of our PCFT problem are:

- *Finding* the bisection bandwidth of a given regular graph; this problem is known to be NP-hard [**?**], with the best known approximation-factor of $O((\log n)^2)$ [**?**].
- Determining an *upper-bound* on the bisection bandwidth of m-regular graphs of size n, for a given m and n. Note that this problem can be reduced to the $k = 1$ version of our PCFT problem. This upper-bound problem has been addressed extensively, and upper-bounds have been determined for small values (upto 4) of m [**?**]; these upper-bound results are not tight.

The above results suggest that, even for $k = 1$, the PCFT problem is likely to be intractable. In general, the PCFT problem can be thought of as the following graph problem: Given n, m, and k (as defined in the formulation), the PCFT problem is to find a k-regular graph over nm-nodes such that the *set* of matchings (i.e., realizable topologies) in the graph maximizes the dynamic bisection andwidth. To the best of our knowledge, the above graph problem (or anything closely related) has not been addressed before.

Proposed Approaches: We will pursue the following approaches for the PCFT problem:

- *Static to Dynamic Conversion.* One reasonable approach to solve the PCFT problem would be to start with constructing a mk-regular graph of n nodes with a high (static) bisection bandwidth, and then group the mk candidate links at each node into m sets of k links each so as to maximize the dynamic bisection bandwidth. For the first step, there are only a few results on explicit construction of general graph classes with high bisection bandwidth [**?**]. In particular, d-regular Ramanujan Graphs [**?**] of are known to have a bisection width of at least $((d/2 - \sqrt{(d-1)})n/2$, but their construction is mostly algebraic. Other graph classes of interest are Cage graphs, which are minimum girth [**?**]. In our specific context of small values (a few hundreds) of km and n, we can investigate bisection bandwidth of certain classes of regular graphs, and pick ones that suggest a high bisection bandwidth with low diameter; in

particular, due to symmetry of nodes/racks, we can also restrict ourselves to "symmetric" regular graph such as distance-transitive graphs. For the second step of grouping candidate links, we will employ certain heuristics. E.g., we can number the n nodes from 1 to n and group the mk links into m sets based on the ranges of node numbers they connect to. Such a heuristic will guarantee a "uniform" division of links into sets across the nodes.

- *Dual-based Approach.* If each rack contains l machines and the links (between a machine and the ToR switch, or a pair of FSOs) have a unit bandwidth, then the optimal desired bisection bandwidth is $nl/2$. We note that inter-rack and inter-machine bisection bandwidths are the same for uniform link bandwidths. Now, let us consider what values of k and m can enable this DBW value of $nl/2$. We consider two extremes: (a) If $k = 1$, then it can be shown that $m = \min(n/2 + l, 7l)$ suffices (but not necessarily optimal). For large values of n and l, it is known [**?**] that $m = 2l$ would almost always work. (b) If k can be an arbitrarily high, then the optimal value of m required is l (for $k = n/2$); here, for $m = l$, the k required (=$n/2$) is also optimal. The above resuls hold for any DBW value that is an integral multiple of n. The above near-optimal solutions for certain special cases of the "dual" problems (i.e., given a desired DBW value, minimize m or k for a given n value) gives us some insights into solving the PCFT and its dual problems. In particular, if we can solve the above dual problem of minimizing m, given k, n, and a desired DBW value, for arbitrary k values and integral of n values of DBW, then it is easy to derive an approximation algorithm for the PCFT problem that has only an *additive* approximation-factor of n.
- *Simulated Annealing.* Simulated Annealing (SA) heuristics [] have been used with great success for optimization problems. To design a simulated annealing approach for our PCFT problem, we need three key components: **(a)** A good "seed" (starting) solution; here, we could use one of our earlier approaches, or as in [**?**], use graphs with a "large spectral gap" [**?**, **?**, **?**] which are known to have desirable properties (e.g., low diameter [**?**]). **(b)** Ways to generate "neighboring" solutions; for this, we can use simple transformations that transform a regular graph to another. E.g., the transformation that changes the edges $(a, b), (c, d)$ to $(a, c), (b, d)$ can be used iteratively to construct any regular graph from another. **(c)** An efficient heuristic for computing DBW of a given graph; for this, we will investigate generalization of the following approaches: (i) Well-known efficient heuristics, viz., SA [] and Kernighan-Lin [] for computing the bisection bandwidth, and (ii) a recent result [**?**] that uses Valiant (or, two-state) load balancing technique [**?**] to compute a *lower bound* on the bisection bandwidth. The above ideas can also be appropriately extended to optimize the traffic-weighted DBW objective.

4.2 Budget-Based Optimization

We now consider the budget-based optimization (*BBO*) problem mentioned at the start of the section. Formally, given the total number of machines to interconnect, physical constraints, overall budget, and pricing of relevant hardware devices, the *BBO* problem is to determine the following such that the dynamic bisection bandwith is maximized: (a) Number of machines (l) per rack and thus, the number of racks (n), (b) number of FSOs (m) per rack and thus, the number of ports on the ToR switch, (b) Number of FSOs ($g \leq m$) that are each equipped with a GM and the number of SMs (k') on each of the remaining $(m - g)$ FSOs, on each rack, and (c) the *pre-orientation* of each of the GMs and *pre-alignment* of each of the SMs in the system.

Task 2: *We will design efficient algorithms for the Budget-Based Optimization Problem (*BBO*) with the objective of maximizing dynamic bisection bandwidth.*

Proposed Approach: Based on our insights from the previous section, we will use the following approaches to address the *BBO* problem:

- *Using PCFT Algorithm.* To use PCFT problem: **(a)** First, we convert the given budget and physical constraints, and the pricing information into a constraint equation over n (number of racks), m (number of FSOs per rack), and k (the number of candidate links per FSO). To relate k to the pricing of SMs and

GMs, note that $km = cg + (m-g)k'$, where c is the number of candidate links a GM can be steered to use and can be assumed to be a constant. **(b)** Second, we solve the PCFT problem for various n, m and k that satisfy the above budget constraint for a given g ($\leq m$). Then, we convert each PCFT solution to a design realizable by g GMs and $(m-g)$ sets of k' SMs each, on each rack, and estimate its DBW using one of the approaches described earlier. **(c)** Lastly, we explore the space of n, m, k, g efficiently using standard search techniques, to compute an efficient network design for the given budget and pricing.

- *Simulated Annealing Approach.* We can modify our Simulated Annealing approach described above for the PCFT problem to solving the *BBO* problem, by appropriately modifying the transformation operator to generate neighbors of a particular network design. Here, the neighbors of a design may include designs with slightly different values of parameters n, m, k, f, and/or candidate links, under the budget constraints.

5 Firefly Network Management

In this section, we focus on the design of a *datacenter management layer* that uses these building blocks to implement a practical reconfigurable datacenter network.

5.1 System Overview

For completeness, we describe the high-level roles of the different components of the management layer in Figure 7.

- **Monitoring Engine (ME):** The input to the management layer is network status information including: (1) link-level information about individual inter-FSO links and (2) measurements of observed traffic patterns or the inter-rack *traffic matrix* as well as views of "elephant" flows [**?**, **?**, **?**].
- **Optimization Engine (OE):** Given the offered traffic workload, the current network state (e.g., active links and link status), constraints on network links defined by pre-configured topologies from Section 4), we need a *topology and traffic engineering strategy* that (near-)optimally meets some desired performance goals (e.g., throughput, loss, or latency).
- **Data plane translation engine (DPE):** The result of the optimization strategy is then translated into the data plane. Here, we leverage recent advances in software-defined networking (SDN) as an enabler for fine-grained control over routing and traffic engineering (TE) strategy [**?**, **?**].
- **Interfaces to users and applications:** Finally, we envision APIs that the management layer exposes to allow users/tenants to leverage the benefits of reconfigurability (e.g., interacting with application-level controllers [**?**, **?**, **?**, **?**]) and receiving hints about the applications (e.g., are they using multipath TCP [**?**]) that can inform the optimization and dataplane modules.

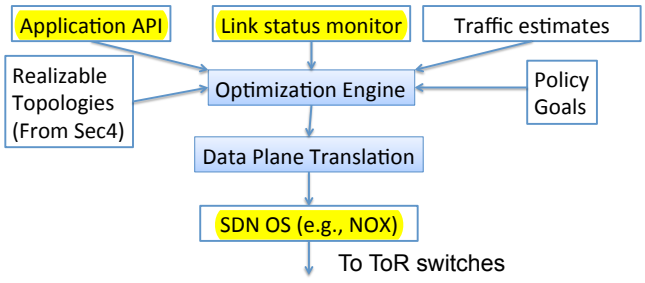


Figure 7: Overview of the Firefly management layer

In designing the Firefly management layer, we build on and extend traditional cloud and network management including traffic engineering [], software-defined networking [], fast routing recovery [], and managing network updates []. The key differences from this prior work arises on three dimensions. First, prior traffic engineering efforts typically assume the topology "as a given", whereas with Firefly, this assumption or constraint no longer applies. This gives rise to new challenges and opportunities for TE and routing in conjunction with *topology engineering*. Second, constraints imposed by the free-space optics (e.g., reconfiguration delay) and pre-configured topologies raise unique challenges w.r.t connectivity and performance guarantees when the topology is in flux which falls outside the scope of prior recovery and configuration schemes. Third, prior SDN proposals assume a "control network" (typically out-of-band) for managing the network devices [**?**, **?**].

In our vision of an "all wireless" fabric, this assumption no longer holds.

Our focus is on the OE, DTE, and the control channel. For the ME, we will leverage past work on scalable traffic matrix and elephant flow detection [**?**, **?**, **?**]. Similarly, we will extend prior work on abstractions for applications to expose their traffic patterns [**?**, **?**]. For clarity, we describe the OE and DTE componentns assuming that we have an out-of-band control network for configuration dissemination and data collection and relax this assumption later.

5.2 Fast reconfiguration and traffic engineering

Task 3: *We will develop fast, near-optimal algorithms for topology reconfiguration and traffic engineering. We will investigate tradeoffs between performance goals using real-world datacenter traces and application requirements. We will also investigate distributed and online algorithms that are robust to estimation errors and do not require global coordination.*

Problem Context and Challenges: At a high-level, we need to solve a a joint topology design and TE problem in contrast to prior work in network management that considers TE in isolation on fixed topologies. We consider a setting with n $Rack$s each equipped with m FSO devices. Let the index i refer to a specific $Rack$ and j refer to a specific FSO device. Thus, the subscript i, j will refer to a specific Rack-FSO combination. Let k denote a pair of racks and let us assume that we know some estimated traffic matrix for every pair of racks denoted by T_k for the volume of traffic from k.

As a starting point, we consider the objective on routing the demands in the traffic matrix to minimize network congestion [**?**, **?**]. There are two main level of control decisions. First, we need to decide if a pair of Rack-FSO links need to be activated; $d_{(i,j \to i',j')}$ is a binary variable that is 1 if we choose to activate $(i, j \to i', j')$. Note that we can only choose from a set of $CandidateLinks_{i,j}$ possible links as defined by the pre-configured topology from Section 4. Second, we need a routing strategy for each demand; let $f_{k,(i,j \to i',j')}$ denote the volume of the inter-rack k demand routed on the edge $(i, j \to i', j')$. Given these control variables, we can model the problem as a Integer Linear Program (not shown) that combines *edge activation* with traditional max-flow like constraints for *flow routing*.

Unfortunately, this problem is theoretically NP-hard (not shown due to space constraints) and it is practically intractable—state-of-art ILP solvers take several hours even for a 20-node problem instance. In practice, we also need to model metrics such as fairness, latency, as well as tenant-provider SLAs. Furthermore, this implicitly ssumes that (a) the traffic demands are known (or predictable) and (b) that the optimization can be run by a global optimizer.

Proposed Approach: We need to a scalable optimization across multiple metrics that is also amenable to a distributed/online implementation. We propose to explore three complementary strategies:

- *Approximation algorithms:* Specifically, we will investigate "randomized rounding" strategies that solve a relaxed linear program (i.e., converting the discrete $d_{(i,j \to i',j')}$ into fractionals) and then selectively choosing to "round" some of these fractions to 1. We can also exploit the "flow" structure of the optimization to design distributed optimization strategies [**?**, **?**].
- *Multi-stage optimization:* Intuitively, solving the traffic engineering problem is simpler than the joint optimization; e.g., using max-flow solvers []. Thus, one heuristic is to see if the new traffic demand can be satisfied by better TE over the existing topology configuration and check if this TE solution is close to the optimal. At first glance, this is a chicken-or-egg problem because we need to run the hard optimization to determine how close to optimal we are! In practice, we only need a bound on the optimal value and not the solution and we can run a relaxed LP to get an upperbound.
- *Exploit real-world structure:* We can leverage the natural structure of real-world workloads; e.g., a small number of "elephant" flows carry the most bytes []. Since these elephant flows are typically long-lived [], they are amenable to coarser time-scale optimizations. Thus, we can design heuristics that use topology

reconfiguration logic for the elephants and fallback to TE for the "mice". This structure is also naturally amenable to an online/local strategy as it avoids the need for global coordination.

5.3 Efficient and consistent data plane strategies

Task 4: *We will design and implement efficient data-plane implementations by extending software-defined networking techniques to guarantee reachability and consistency properties in the presence of reconfiguration and link dynamics.*

Problem context: Our focus here is on translating the solution provided by optimization engine into a practical data plane forwarding strategy. Here, we leverage recent advances in software-defined networking (SDN) to implement the topology and routing reconfiguration strategies []. While SDN is an "enabler" as it provides cleaner management abstractions and open interfaces (e.g., via APIs such as OpenFlow []), Firefly introduces unique efficiency, correctness, and consistency challenges during network reconfigurations and local link readjustments.

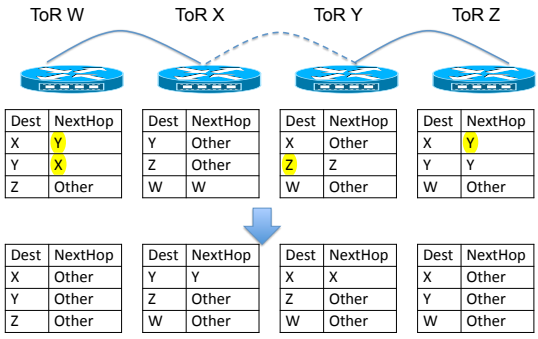


Figure 8: Example to illustrate the need for careful and consistent routing management in Firefly

Network reconfigurations: Consider the scenario in Figure 8, where Firefly is reconfiguring the network topology and we want to active the X-Y link and disable X-W and Y-Z. The problem here is that during the reconfiguration, tables of the switches may have an inconsistent view of the topology. For instance, if we update the forwarding tables to use the X-Y link before the link is active or conversely if we do not update the routes to disable routes using Y-Z or X-W, then we may have connectivity issues. Furthermore, if there were large elephant flows using the Y-Z or X-W links, then moving them may cause transient congestion in other parts of the network and impact their completion times.

Our goal is to ensure that even during this transient periods, packets are not dropped; we get reasonable performance; and there are no adverse effects due to forwarding loops. While this seems related to work on *consistent updates*[4] in SDN [**?**, **?**], this is indeed a useful theoretical framework, there are two unique challenges. First, this prior work implicitly assumes that the network topology has not changed between the two update states—in Firefly, the topology might itself may change. Second, it does not provide guarantees on the bandwidth or the throughput during the transition period [**?**].

Link flux: Second, even with a static topology state, links may be transiently unavailable because of the "micro alignment" that each FSO device may need to run. The timescales of such micro-alignment may be much smaller than the time needed to communicate with an SDN controller and waiting for rule updates [**?**]. In fact, it may be counterproductive to report such transient link failures to the controller as it may cause needless topology reconfigurations. Thus, in addition to the link micro-alignment mechanisms discussed in Section 3, we also need corresponding network layer techniques.

Proposed approach: We believe it is useful to decouple the three types of requirements: *connectivity during reconfigurations*, *connectivity during link realignment*, and *performance* as these entail different solution strategies. Corresponding to these requirements, we will explore three techniques:

- *Guaranteeing reachability during reconfigurations:* We will explore approaches to guarantee end-to-end reachability even in the presence of topology reconfigurations. For instance, we will set aside a subset of the $k$links to be "always on" (or statically aligned) to create a connected "backbone" (e.g., a spanning

[4]This guarantees that each packet is either processed by an old configuration or a new configuration but not a mix.

tree) to ensure that all pairs of racks can always reach other. By itself, a backbone does not guarantee data plane reachability, and we need to *carefully schedule* the reconfiguration operations.

Consider the earlier example from Figure 8. The key here is in the ordering of the steps—we remove routes before deactivating links and add routes only after activation is complete in the following sequence: (1) Update routing table to reflect the removal of links (x, w) and (y, z); (2) Switch the FSO links to activate/deactivate links; and (3) When (2) is complete, we update routing tables to reflect the addition of link (x, y). We believe this intuition can be extended to the multiple reconfiguration case as well by a combination of suitable batching and parallelization strategies (e.g., find non-conflicting scenarios.)

- *Handle link flux locally:* Future SDN roadmaps have provisions for local recovery mechanisms analogous to similar schemes in the MPLS and SONET literature []. We will explore the available alternatives for our prototype implementation. In the absence of such features, we will explore the possibility of running a local "lightweight" SDN controller on every rack that can quickly react to the micro-alignment changes but relies on the global controller for longer-timescale reconfigurations [**?**].
- *Minimizing impact of reconfigurations:* In order to minimize the impact of a reconfiguration on network congestion, we can extend the optimization from **??** and additional objective criteria (or constraints) that will force the optimizer to prefer reconfigurations that cause minimal disruption for the active "elephant" demands. Because Firefly has more degrees of topological freedom relative to prior work, we may be able to minimize disruptions for existing connections by simply introducing additional links on-demand.

5.4 A wire-free control channel

Task 5: *We will design and implement a RF-based protocol that will provide robust control channel for Firefly.*

Problem context and challenges: Existing work in the SDN-style centralized network management literature either implicitly or explicitly assumes the availability of an "out-of-band" control channel that is not managed by the SDN network itself []. This control channel is typically used for the controller-switch protocols—delivering configuration commands and collecting switch statistics. Otherwise, there can be subtle bootstrapping problems w.r.t availability of the control channel itself.

As discussed in the previous section, we may have to engineer some level of reliability/consistency mechanism even for the regular inter-rack fabric. We can also exploit this basic reachability framework as a basis for in-band control. For instance, we can setup static shortest paths between each FSO switch to the controller and not reconfigure them. That said, we still have a bootstrapping problem where these switches will need to discover paths to the SDN controller. Furthermore, there is still a concern that the control path may be transiently unavailable during micro-alignment pauses. While these are not fundamentally intractable, we want a highly reliable and low-latency control channel.

Proposed Approach: A promising alternative to in-band control is to equip each ToR switch with a lightweight commodity RF interface. Because the bandwidth requirements of this control channel are typically not that high, we believe we can use a simple RF-based wireless control channel for the entire datacenter. Consider two cases. First, even if we send 1000 configuration commands per ToR switch per second, the total bandwidth requirement will be less than **10 Mbps** per-rack. Second, even if we want to collect per-flow statistics per-second from every ToR switch assuming roughly 10K flows/second per-rack and assuming a 100 byte flow record size we will only need **10 Mbps** per rack.[5]

The more critical challenge here is *latency* of the control channel especially for configuration commands. Specifically, if the control loop delay is too high then it might induce some stability problems for

[5]Since the bandwidth demands are low, we could engineer a out-of-band channel with a few switches as well. However, this goes against our overall vision of a pure wireless network fabric and thus we plan to investigate eliminating this wired control fabric as well.

the reconfiguration algorithms as they may not be able to converge in reasonable timescales. (Note that we can tolerate some error or delay in the data collection or correct for it in the reconfiguration algorithms described earlier.) Unfortunately, existing "commodity" wireless MAC protocols are not geared toward such low-latency.

Because we only need one of these devices per rack, we will design a custom software-radio based solution using **XXX Samir please fill**

6 Extensions to Architecture

In this section, we discuss certain extensions to our architecture that we would investigate in our research.

- *Non-ToR Switches and FSOs.* As described before, our network architecture consists of only ToR switches whose ports are connected to the rack machines or the FSOs placed on the rack. Incorporating non-ToR switches (as in most data center architectures []) can add more flexibility to our design. The challenges would be: (a) Finding sufficient physical space to place the FSOs connected to such non-ToR switches, and (b) solving the PCFT problem would also entail determining the interconnections between these switches, and the PCFT solutions would involve more general non-regular graph.

- *Dynamic Reconfiguration of Link Bandwidths.* In our design, the bandwidth of each FSO link is limited by the capacity of the ToR switch port, since each FSO connects to a port on the switch. One way to embed more flexibility in our network design is to facilitate variable bandwidth FSO links. This can be achieved by having each port of ToR associated with a unique wavelength, and using a multiplexer and a WSS (wavelength selective switch) unit between the ToR switch and the FSOs, as in OSA [**?**]. In essence, the multiplexor and WSS allow one or more ToR ports to "feed" into a single FSO link, and hence enabling variable bandwidth FSO links. The WSS can be configured in real-time to yield variable and reconfigurable bandwidths to FSO links. It would be interesting and challenging to incorporate the above dimension of flexibility into our network design and generalize our PCFT, BBO, and reconfiguration algorithms.

- *Multicast.* Big data applications have diverse communication patterns that mix together unicast, multicast, all-to-all cast, etc. Recent works show [**?**] show that all-to-all data exchange on average accounts for 33% of the runnign time of Hadoop jobs. As suggested in [**?**], the optical communications are particularly amenable to efficient implementation of such *-cast patterns by leveraging various components such as directional couplers, wavelength-division multiplexed, etc. Incorporating the above ideas in our design will require making challenging design choices.

- *Vertically Steerable FSOs;* 45° *Mirror Poles.* In our design, we use a ceiling mirror to circumvent physical obstruction for line-of-sight FSO communication. However, in certain contexts such as outdoor scenarios for containerized architectures [**?**] installing a ceiling mirror may not be feasible. In such cases, we need other mechanisms for line-of-sight communications. E.g., we can install FSOs on vertically-steerable poles such that each link operates on a separate horizontal plane. To avoid physical obstruction due to the poles, shorter distance links can be operated on a lower horizontal plane than the longer distance links. Another possible mechanism could be to have FSOs direct their beams to vertically-steerable small mirrors angled at 45°.

7 Prototyping and Evaluation Plan

Task 6: *To demonstrate the viability and benefits of our proposed vision, we will evaluate our approach both at an individual component granularity as well as an end-to-end prototype and testbed demonstration.*

- **Design and protoype compact, cost-effective, steerable FSO devices for datacenter-scale use:** We will prototype proof-of-concept SFP-based FSO devices with a small form factor that provide a high degree of flexibility as outlined in Section **??**. We will design optical mechanisms to keep the laser beam collimated over a range of distances (few meters to about 100m) using 10G optical SFPs at the end points. We will also prototype and evaluate two different steering mechanisms: using switchable mirrors and galvo motors. The designs of the collimator and the steering mechanisms will proceed independently at first as we already have access to commodity FSO platforms [] (that are used for outdoor, long-distance communication) that can be repurposed for initial development of the steering mechanisms.
- **Reliability of steerable FSO in realistic datacenter conditions:** Real datacenters will likely have several sources of "disturbances" (e.g., vibrations of racks, temperature gradients, airflow patterns, etc.) that may cause alignment and performance issues for FSO-based link. To address this concern, we will use a two-pronged approach. First, we will create a lab environment that can emulate the effects of different types of disturbances. To inform the scope of these effects we will actively engage our industry partners as well as adding lightweight instrumentation sensors to compute clusters at local organizations (e.g., Brookhaven National Lab and CEWIT. Second, we will deploy a small number of FSO links inside an actual datacenter environment that we already have access to (CEWIT in Stony Brook University) and conduct a longitudinal study of the reliability of the links under various environmental factors. The final goal of this stage is fine-tuning the FSO design via systematic stress-testing and pick one steering design for the end-to-end protoyping and evaluation (described momentarily).

  [SD: do we have access to a lab that can emulate "disturbances"??]
- **Performance and power benefits under realistic workloads:** We will develop scalable packet- and flow-level simulation platforms building on prior work [**?**, **?**] to evaluate the potential benefits of our topology design algorithms (Section 4) and reconfiguration algorithms (Section 5). We will start with publicly available workloads from previous studies and extrapolate them to evaluate the effectiveness of our proposed algorithms [**?**, **?**, **?**, **?**]. We will also actively work with industry supporters (e.g., see letters from Facebook and Microsoft) to evaluate the benefits of our concepts using real traces.
- **Scalability, responsiveness, and correctness of control plane:** We will begin by implementing a proof-of-concept controller in academic SDN control software such as POX [**?**] and as the project matures we will also port to open-source commercial platforms such as OpenDayLight that are more widely used []. we will create benchmark suites to "stress-test" the scalability and responsiveness of the control plane reconfiguration algorithms. We will use emulation platforms such as MiniNet [] and Emulab [] to emulate failure and reconfiguration scenarios to test the correctness of the proposed local recovery and consistent reconfiguration mechanisms.
- **End-to-end integration and evaluation:** A full-scale datacenter deployment and testbed is outside the scope of the proposal both in terms of infrastructure and personnel resources.[6] Within the scope of our budget, we plan to demonstrate a proof-of-concept testbed of 4 nodes (node represents a rack). Each node will be essentially a NetFPGA card [] on a host computer. Each NetFPGA card has 4 x 10G SFP ports, three of which will connect to a FSO device each with one left for the controller use. We will use openflow switch implementation on the NetFPGA cards [] to represent the ToR switch. Using NetFPGA will allow us use of precise traffic generators for repeatable traffic loads [], perform very fine-grain, per-hop timing measurements and performance analysis in the openflow switch, and easy access to the diagnostics information (DOM or digital optical monitoring []) from the optical SFP for alignment/steering and as well as link characterization. It will also perhaps allow us to study innovative packet hadling/forwarding not possible with commodity SDN-capable switches.

  The 4 node setup (along with the 4x3=12 FSO devices) will be deployed on top of the racks in

[6] We plan to develop separate infrastructure grant proposals to develop at-scale prototypes.

an operational data center (in CEWIT) for testing in realistic environments. The nodes will be moved around on different racks to create various geometric possibilities. This will create various stress cases for studying the stability of the FSO link and steering performance. [SD: will somebody complain that real data centers have real obstructions so such deployment is difficult?] In addition to the characterizing the raw performance of the links, a variety of synthetic and trace-driven traffic load will be used to evaluate the end-to-end, application perceived performance of the entire system.

8 Education and Broader Impact

Some input from Jon would be helpful too.

Broader Impact. Performance of data centers, energy savings (energy proportionals DCs), cabling complexity, broaden the current applications of FSO communication (which is currently limited to specialize across-town applications), ad hoc deployment of interconnection architectures (e.g, for Helios type containerized DCs),

Integration of Research and Education Multi-disciplinary (CS and Mech). Will develop multi-disciplinary courses (online too).

Engaging High School and Undergraduate Students. Long Island have some of the best public schools in the country and we are keen on tapping into this high school talent. SUNY has a Simons Summer Research Program[7] that provides a mechanism to recruit talented high school students. During the past few summers at SBU, our colleagues have also organized Engineering Camps to attract high school students to come to SBU; At the camps, the students have several two-day laboratories in which they are instructed on how to design, build, and program various types of devices. The above programs would provide perfect avenues to recruit a few high-school students for summer projects related to our research. The PIs in the past have also used REU supplements to their NSF awards to engage undergrads in their research, and would continue to involve undergraduates in our research.

Due to the hi-tech appeal of free-space optics in data communications and other applications, we are keen on giving presentations and demonstrating appropriate aspects of our research prototype to some high-schools and our undergraduate studnets. We believe that the obvious appeal of free-space optics and steering mechanisms will be exciting for the students, and give us an opportunity to further encourage and recruit some of the best students. Finally, we plan to build "kits" that can be used by the students to build hobby projects, e.g., FSO-based scanning devices, inexpensive custom-built steering mechanisms for FSO devices, demonstration of high-bandwidth FSO links using commodity hardware, etc. More elaborate projects based on the above ideas would be ideal for our CS Honors senior projects. We hope to motivate them high-school students to pursue further education and careers in computer science. Many of the Simons Summer Research Program participants have excelled at the Intel Science Talent Competition (ISTC), and we are keen on mentoring high school students for ISTC.

Involving Under-Represented Groups. SUNY Stony Brook has a history of active outreach efforts in order to involve traditionally under-represented groups in science and engineering research. SUNY Stony Brook has the Turner Fellowship Program for under-represented groups, the SUNY Alliance for Minority Participation (SUNY AMP), a minority faculty recruitment initiative, and the SUNY Alliance for Inclusive Graduate Education and the Professoriate (SUNY AGEP). Research in undergraduate studies will also be integrated through the *Women In Science & Engineering (WISE)* mentoring program in SBU, which regularly offers four-week research and inquiry-based courses. We plan to introduce a new WISE course related to free-space optics communications and applications. The PIs are committed to involve under-represented groups in "high-tech" research and development.

[7] http://www.stonybrook.edu/simons/

9 Results From Prior NSF Support

Samir R. Das and **Himanshu Gupta** are PI/Co-PIs on the following recently concluded/ongoing NSF awards: i) 'A Market-Driven Approach to Dynamic Spectrum Sharing' (2008-13, $406,000), and ii) 'Understanding Traffic Dynamics in Cellular Data Networks and Applications to Resource Management,' (2011-14, $320,425). These projects focus on developing market-driven algorithms and systems for dynamic spectrum access systems (first) and understanding spatio-temporal traffic dynamics in cellular data networks via analysis of network traces and using them for spectrum/energy management applications (second). Over 15 papers were co-authored by the PIs related to these awards and 6 PhD students received direct support. The PIs gave several public lectures based on the results. [SD: if we have space we may also mention the sensor grants.].

Vyas Sekar is a PI on two recently awarded NSF grants "Enabling Flexible Middlebox Processing in the Cloud" and "Rethinking Security in the Era of Cloud Computing" starting in Sep 2013. The research proposed therein focuses largely on "middlebox" functionality such as IDS, firewall, and proxies and does not focus on the datacenter topology and routing aspects. These projects have just commenced and there are no outputs at this time. As such the proposed research in these projects does not overlap with the management layer/SDN approaches proposed here.

References

[1] `http://www.fsona.com/product.php?sec=2500e`.

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

[3] A. Curtis et al. Legup: using heterogeneity to reduce the cost of data center network upgrades. In *Co-NEXT*, 2010.

[4] A. Greenberg et al. Vl2: a scalable and flexible data center network. In *SIGCOMM*, 2009.

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

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

[7] J. Shin et al. On the feasibility of completely wireless datacenters. In *ANCS*, 2012.

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

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

[10] D. Kedar and S. Arnon. Urban optical wireless communication networks: the main challenges and possible solutions. *IEEE Communications Magazine*, 42(5), 2004.

[11] Jayaram Mudigonda, Praveen Yalagandula, and Jeffrey C. Mogul. Taming the Flying Cable Monster: A Topology Design and Optimization Framework for Data-Center Networks. In *Proc. USENIX ATC*, 2011.

[12] Luka Mustafa and Benn Thomsen. Reintroducing free-space optical technology to community wireless networks. In *Proc. 19th Americas Conference on Information Systems, Chicago, August, 2013.*, 2013.

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

[14] L. Popa. A cost comparison of datacenter network architectures. In *Co-NEXT*, 2010.

[15] Ankit Singla et al. Proteus: a topology malleable data center network. In *HotNets*, 2010.

[16] Ankit Singla, Chi-Yao Hong, Lucian Popa, and P. Brighten Godfrey. Jellyfish: Networking data centers randomly. In *NSDI*, 2012.