1 Introduction

Data centers (DCs) are a critical piece of today's networked applications in both the private [] as well as the public sector []. The key factors that have driven this trend are the economies of scale, reduced management costs, better utilization of hardware via statistical multiplexing, and the ability to elastically scale applications in response to changing workload patterns.

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

Meeting these multitude of requirements is as critical to the computing ecosystem as it is challenging. Traditional DC network architectures can be broadly divided into two categories: (1) *overprovisioned* fabrics (e.g., fat-trees or multi-stage Clos networks) with full-bisection bandwidth [] and (2) *oversubscribed* fabrics (e.g., '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 somewhat extreme 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 are promising, existing augmentation approaches are incremental and only offer limited flexibility; e.g., optical augmentation is effective only for simple workloads that are amenable to bipartite matchings between top-of-rack switches [] and wireless augmentation is limited by interference/range constraints []. Furthermore, all of these architectures incur high cabling cost and complexity [**?**]. (We elaborate on these factors in Section **??**.)

Our vision: In this proposed research, we consider an *extreme* design point. Instead of trying to incrementally improve the poor cost-performance tradeoffs, high cabling complexity, and limited flexibility of the existing DC architectures, we envision a *flexible*, *all-wireless* inter-rack fabric.

Figure 1 shows a conceptual overview of our vision called Firefly.[1] Each top-of-rack (ToR) switch is provisioned with reconfigurable wireless links that can reach a subset of other racks. The datacenter management layer reconfigures the network topology to adapt to current traffic workloads. Our insight here is that topological flexibility (if done right) can replace the need for overprovisioning. Wireless naturally eliminates the cabling complexity and attendant operational overheads (e.g., obstructed cooling) [**?**], and facilitate new topologies that would otherwise remain "paper designs" due to cabling complexity [16]. Furthermore, flexibility can reduce energy costs [**?**, **?**] and enables incremental expandability [16].

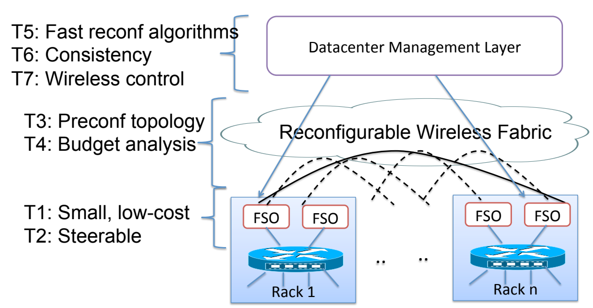


Figure 1: Overview of the Firefly vision

Research plan and Intellectual Merit: To realize the all-wireless vision outlined above, however, we need to look beyond traditional radio-frequency (RF) based (e.g., 60GHz) wireless solutions as they are fundamentally constrained in terms of range, capacity, and interference. To this end, we rely on *Free-Space Optical communications* (FSO) as it can offer very high data rates (tens of Gbps), with long range (100m), with low transmission power, and with low interference footprint.

The three characteristics of our approach—FSO-based inter-rack links, all-wireless, and topology flexibility—raises unique algorithmic, networking, and system design challenges along three thrusts (Figure 1):

[1] Firefly stands for XXXFillmeXXX.

- 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 develop cost-effective 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, the physical and geometric constraints of the steering mechanisms raise new challenges for architecting DC subject to budget constraints (**??**).
- Our vision imposes new network management challenges and requires novel algorithms for joint topology and traffic engineering (**Task 5**, new consistency abstractions that guarantee reachability and performance when links may be in flux (**Task 6**). Furthermore, we need new wireless mechanisms to replace traditional wired control channels (**Task 7**).

Team Qualifications Our team comprising 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 []—is uniquely positioned to tackle the aforementioned challenges. Our proposed research is highly integrative and the PIs expertise complement each other. The PIs have an established history of collaboration [] and outreach activities [], and this research will further strengthen these.

2 Motivation and Research Overview

As our title suggests, [SD: best to refer to intro than title] there are three key aspects to our vision: *flexibility*, *wireless*, and the use of *free-space optics* in DC networks. We begin by arguing why each of these aspects is needed before providing a high-level view of our proposed architecture.

2.1 Case for Flexibility in Datacenters

Our focus in this paper is on the inter-rack fabric connecting different top-of-rack (ToR) switches. At a high level, we can classify existing designs along two axes: (1) the extent of oversubscription and (2) flexibility (if any) to reconfigure links.

Table **??** summarizes prior work along these two key dimensions. Traditional static topologies such as leaf-spine fabrics [] or fat-tree architectures [] represent extremes in the space of cost-performance tradeoff. Additionally, such structured graphs impact incremental expansion [16]. Recent measurements show that the DC traffic patterns exhibit hotspots of inter-rack activity with a few "heavy" flows [**?**, 4, 5]. These motivated designs where an oversubscribed core is *augmented* with a *few* flexible optical connections [] or wireless links []. However, these are limited in the degree of flexibility and introduce additional challenges. Optical solutions create a single point of failure, and cannot handle other one-to-many or many-to-one demand patterns [5]. Wireless links, on the other hand, are fundamentally limited in the capacity and range; e.g., even recent solutions cannot provide more than XXXFillmeXXXMbps or XXXFillmeXXXmeters. Finally, they inherit the cost and cabling complexity of the wired "core" they seek to extend.

Category	Backbone	Flexibility	Notes
Leaf-Spine (e.g., [])	Wired, over-subscribed	None	Poor performance
Full bisection bandwidth (e.g., [2,4])	Wired, no over-subscription	None	High cost + cabling complexity, no incremental expandability
Wireless augmentation (e.g., [**?**, 8])	Wired, over-subscribed	Few 60Ghz links	Low range, bandwidth
Optical augmentation (e.g., [**?**, **?**])	Wired, over-subscribed	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

Rather than incrementally improve an oversubscribed network, we posit that flexibility, if designed suitably, can obviate the need for overprovisioning and the need for a static backbone! This can provide a dramatically improved point in the cost-performance tradeoff. Furthermore, this flexibility can enable energy savings by selectively shutting down links depending on the 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.

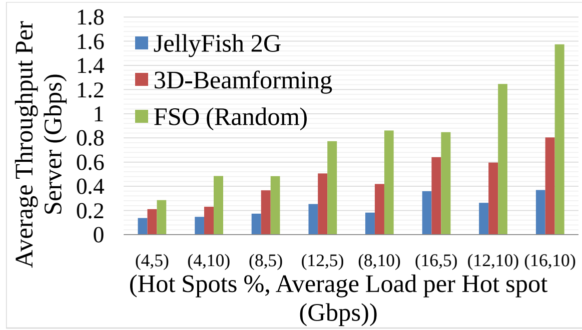


Figure 2: The case for flexibility – it can provide performance comparable to a full bisection bandwidth network with a lot less equipment and even get rid of aggregation layers. PLACEHOLDER from hotnets

Figure 2 shows that by increasing the degree of flexibily XXXFillmeXXX.

2.2 Case for Wireless via Free-Space Optical Communications

Why wireless? To realize such a flexible fabric, conceptually we need a reconfigurable "patch-panel" between pairs of racks [?]. Of course, such a big-switch abstraction is infeasible: (1) it requires very high fanout ($n \times D$, where n is the number of racks and D is the number of flexible links at each ToR) and backplane switching capacity; and (2) the cabling complexity would be prohibitive and create operational overheads w.r.t failures, and cooling/airflow considerations []. For similar reasons, traditional optical switching is also not viable. Finally, this giant switch introduces a single-point of failure [?, ?, 15]. To avoid the need for such a massive switch, we turn to *reconfigurable wireless* links between the ToR switches.[2]

Why FSO? The seemingly natural solution then is traditional radio-frequency (RF) wireless technologies (e.g., 60GHz). Unfortunately, these have many fundamental performance limitations: (1) RF links produce a large interference footprint due to a wide beamwidth, even with new "ceiling mirror" architectures [8]; [SD: This blames one paper only. This does not argue why we are discounting every form of RF. Recall the comment from hotnets reviewer. I can fix this. But the real argument is somewhat involved – more than a oneliner.] (2) The beam-steering technologies to implement flexibility are slow and inaccurate [8] and increase the interference footprint [?]; [SD: Need to rephrase, overly sweeping.] and (3) The data rates of RF links fall off rapidly with distance [8] and the use of higher transmit power to increase the range will increse interference and is limited by regulations []. To overcome these limitations, we leverage a somewhat non-standard wireless technology—free-space optics (FSO) that uses modulated visible or infrared (IR) laser beams [10].[3] We elaborate on the advantages of FSO vs. RF in Section 3.

2.3 Architecture and Proposed Research

[2]Note that we are not proposing a fully wireless data center [7]; our focus is on the "inter-rack" fabric.

[3]Unlike traditional optical links, the laser beam in FSO is not enclosed in a glass fiber, but transmitted through the air (and hence "free space").

Combining the above arguments leads us to the architecture in Figure 1. We eliminate the need for a wired backbone network and rely on a reconfigurable FSO-based wireless fabric. Each ToR switch is equipped with a pre-specified number of FSO devices and each FSO device assembly is capable of precise/fast steering to connect to target ToRs. To establish an *obstruction-free* optical path, the space above the racks is a natural choice for laser propagation as this space is roughly free from obstruction [**?**]. The FSO transceivers will be anchored on the top of the rack and connected to the ToR switch. To ensure that the devices themselves do not obstruct each other; we propose to vertically stagger them or use ceiling mirrors (and possibly additional mirrors on the beam path) [].

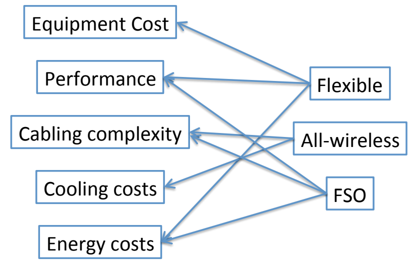


Figure 3: Overview of the key concepts underlying Firefly and how they benefit different aspects of datacenter considerations from Section 1

The DC management layer intelligently reconfigures these devices to adapt to changing network requirements. Figure 3 summarizes how the three key aspects of Firefly—flexibility, all-wireless, and use of FSOs— benefit different considerations of DC network design: (1) Flexibility ensures high-performance with lower cost and enables energy reduction []; (2) A wireless fabric eliminates concerns about cabling complexity and interference with cooling [**?**]; and (3) Using FSOs eliminate performance concerns for a wireless network that might arise from range and interference constraints.

With this context, we discuss the three broad research thrusts we need to address to turn the benefits (Figure 2,3) into reality: (1) **feasibility of FSOs for Firefly (Section 3)**; (2) **foundations of flexible topology design (Section 4)**; and (3) **effective datacenter management (Section 5).**

[VS: revisit these bullets to be consistent with intro of each section]

3 Designing FSO Links for Flexible Inter-Rack Networking

[SD: terminology question: FSO link, transceiver, device, device assembly, or system?? Among these device or system sound too generic and something a non-technical person would use. We can use link and transceiver depending on context. They are also more common in networking lit.]

In this section, we begin by specifying the design requirements for FSOs and highlight why existing FSO technologies fail to meet the requirements that the Firefly vision imposes. Then, we highlight a design roadmap for meeting these requirements.

3.1 Overview and Requirements

The design of FSO transceivers in Firefly must simultaneously meet the following requirements:

- *Size, power, and cost effectiveness:* Our goal is to design a single FSO transceiver assembly (i.e., including alignment and beam redirection machinery) will have ≈ 3"x8" footprint so that a few tens of such devices can be packed on the ToR. The power consumption should be modest and they must be cost-competitive to existing networks.
- *Ability to provide 10-100Gbps data rate:* As DC traffic rates are growing [**?**] and demands for 40 Gbps networks emerge, our design must be capable of providing high throughput.
- *Fast and precise alignment and steering :* [SD: terminology question: 'steering' or 'redirection' here?? note that SM is not technically steering. So we were looking for more a general word. Redirection is a better word as then we can say switching for SM and steering for galvo. Steering brings in the sense that it is continuous, but it does not need to be get to what we want.] For FSO links to provide high throughput, the transmit/receive devices must be precisely aligned. Thus, we need mechanisms for robust re-aligment in the presence of environmental effects; e.g., vibrations, changes in airflow. Furthermore, to provide fine-grained reconfigurability, we need to be able to steer the laser beams to connect to another FSO transceiver on another rack determined by the management layer in Section **??**.

Unfortunately, existing FSO transceivers target *fixed* terrestrial long distance (miles) communication [] and do not meet our size, power or cost goals. For example, a typical commercial system [**?**] is 2 cubic feet, costs $5-10K for a single link, consumes XXXFillmeXXXwatts. The reason for the significantly large numbers is that they have to overcome outdoor challenges — beam path variations due to a host of environmental factors [], larger transmit power requirements for longer distances; alignment problems due to structural swaying etc. While these issues largely disappear in the context of DCs and thus create a pathway for size, power, and cost-optimized design, we have a new requirement of fast reconfiguration.

We outline the challenges and proposed approaches for FSO design in Firefly in a two-step process: (1) Designing the basic FSO link for a DC-scale operation, and (2) Fast and precise beam redirection to enable reconfiguration. Our research here will inform the parameters (e.g., range, size, cost) which will be input to the topology design problems in Section 4.

3.2 Cost-effective, Small-Form Factor, and High Throughput FSO Links

Task 1: *We will design the FSO link including transmitter, receiver and the optical beam path for effective datacenter scale operation, including the size, power and throughput requirements.*

[SD: I modified the task a little. Pls review.]

An FSO communication link has three basic components: (i) a modulated laser source (typically infrared) as a transmitter (TX); (ii) a high-speed optical detector/demodulator receiver (RX); and (iii) a robust optical path between TX and RX. As a cost-effective, compact, and commoditizable solution, we will leverage optical small form-factor pluggable (SFP) transceivers [] for TX and RX components, instead of a first-principles design. Optical SFPs are widely used to interface optical fibers with (electrical) packet switches, are small (XXXFillmeXXX), and do not create an additional power burden.[4] Further, using them makes it easier to 'ride the technology curve' – the data rate supported in the FSOs will be no worse than supported in commodity optical fiber-based communications. [SD: We never say anything about data rate. So we now said it here.]

[SD: The rest of this subsection is rewritten with more focus.]

Divergence In contrast to standard 'wired' optical links where the laser beam from the optical SFP launch directly into the fiber, the optical path in our design is established by launching the laser beam into free space. A fundamental optical property (unrelated to use of SFPs) is that the laser beam always *diverges* in a cone as it propagates in free space and accordingly the power density on the transverse plane goes down with distance. To minimize this divergence, we need to design suitable *collimation lens solutions* on the optical path near the TX that makes the laser beams roughly parallel (diverging very slowly with an angle in order of milli-radians, say). A similar lens near the RX will focus the beam back onto the detector. While the general idea of using lenses to reduce divergence is well known in optics and more specific solutions in the context of conversion of optical SFPs for FSO communication have been pursued before [12], this brings in new design challenges in DCs. We articulate them below.

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 TX) and the rate at which it diverges beyond this point (divergence angle). To keep the divergence minimal, the beam waist must be large requiring a larger lens with a longer focal length, in turn increasing the distance between the SFP and the lens. This increases the size of the assembly – a concern for Firefly. On the other hand, a smaller waist may address the size issue, but will cause the beam to diverge too quickly with the power density at the detector falling behind the 'detection threshold,' especially for the longer links. This requires a careful balance in the design. Our initial calculations (not shown here due to lack of space) show that achieving this balance is indeed possible as the optical SFPs employed for long distance fiber communications are very sensitive (very low detection

[4] They will likely be used for high datarate links in DCs in any case.

threshold). Also, it is possible to accommodate a long the optical path between the SFP and the lens within a small space by reflecting it multiple times with small mirrors (similar idea used in Figure 5).

Alignment Alignment presents a somewhat related, but critical challenge. Considering the transverse plane, the beam power falls off from the beam center following a Gaussian profile []. Obviously, everything else remaining equal there is a loss of received power if the RX is off center. However, the Firefly design must be forgiving for small natural shifts of the optical path during regular DC operation (e.g., due to rack vibrations or drifts due to temperature variations). This again calls for a larger diameter waist bringing in the same challenges that we just described.

Regardless the above, the beam must be re-aligned occasionally to correct for unexpected shifts and also at the time of pre-configuration (described momentarily). We propose to use piezoelectric positioners or thermally expandable materials to provide fine-grained adjustment to re-align the RX detector at the "peak" energy position. Commodity technologies are available to develop these solutions []. The feedback needed for the correction can be obtained from the DOM (digital optical monitoring) support available in the optical SFP standard and carried on the I2C bus via the connectors on the SFP [].[5]

In summary, we will (a) demonstrate the viability FSO communications using commodity components that are size, power and cost efficient. (b) investigate the size-performance tradeoff in a DC-specific context and (c) design robust techniques for alignment adjustment. [SD: We haven't argued anything about the cost much except saying that these are commodity. SFP may be of low cost, but not peizo-positioners.]

3.3 Precise and Fast Beam Redirection

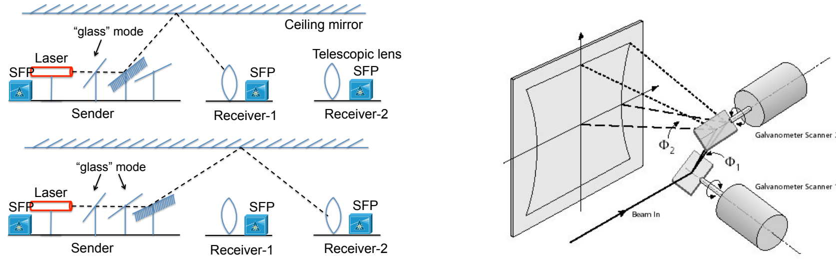
Task 2: *Develop fast and effective beam path redirection techniques to achieve reconfiguration in the interconnection fabric.*

The result of the previous investigation will provide the basis for a high-speed reliable link but by itself it offers no *flexibility* to reconfigure the links. To this end, we need efficient *beam steering solutions*. ~~We have qualitatively investigated a wide spectrum of candidate solutions including XXXFillmeXXX, XXXFillmeXXX, XXXFillmeXXX. Unfortunately, these fail to meet one or more of our cost, speed, or commoditizability needs. For instance XXXFillmeXXXprovides XXXFillmeXXX.~~ [SD: I am not sure that we have examples like this. So, I add a more modest statement below.] A wide spectrum of candidate solutions exist for laser beam steering including phased array techniques []. But most of these are not commodity and some are subjects of active research. Feasibility and cost-effectiveness for adapting these solutions for DCs are unknown. For feasibility reasons, we have investigated two candidate commodity technologies that we will describe in this section. Irrespective of the technology used, there are two fundamental granularities of beam "movements":

1. *Steering:* This refers to the central mechanism where a FSO beam emanating from a TX is redirected to a different RX. This must be done at a fast time scale – few millseconds in order to be responsive to DC traffic dynamics. Physical and optical limitations, however, induce constraints such that each transceiver may only be able to link to only a *subset* of other transceivers in the DC. The specific types of constraints may be technology-specific as we will see later.
2. *Preconfiguration:* Given the above constraint, the above subset has to be chosen in a semi-offline fashion for each FSO transceiver independently. This gives rise to interesting topology design problems that we address in Section 4.

In our proposed research, we will investigate two promising solution strategies for Steering with different tradeoffs discussed below.

[5] We suspect for most effects, corrections on the RX side are sufficient. If TX side needs to be adjusted as well, the RF-based control channel from Section 7 can be used to coordinate the alignment on both ends.



(a) SM: In the top-half the second SM is in mirror mode, redirecting the beam to Receiver 1, while the bottom half has SM3 in mirror mode and thus redirecting to Receiver 2.

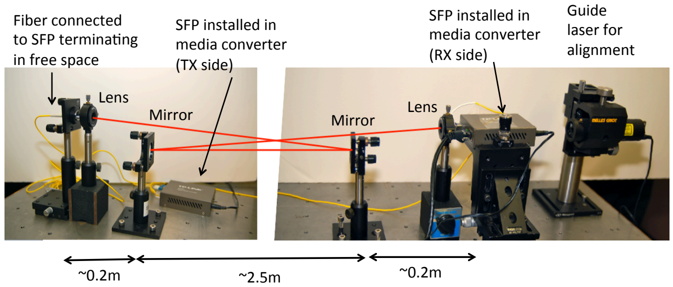
(b) GM: Two mirrors direct incident beam into a rectangular cone.

Figure 4: Candidate beam re-direction approaches.

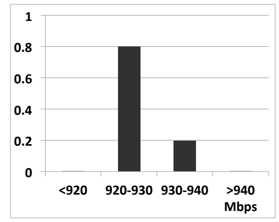
Switchable Mirrors Switchable mirrors (SMs) are made from a special liquid crystal material that can be electrically controlled to rapidly switch between reflection (mirror) and transparent (glass) states at millisecond timescales [**?**]. These are used in various forms of visual aids in niche markets e.g., rear-view video mirror. Figure 4(a) conceptually shows how we can use SMs for beam redirection. Each FSO device will be equipped with multiple SMs, with each SM pre-aligned (offline) to a dedicated beam path (part of Preconfiguration). The desired link is established by swithcing 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, but not shown for ease of visualization.) As mentioned earlier, the ceiling mirror redirects the beam back to the receiving rack, making efficient use of the above-rack space while minimizing interference. [VS: cost?]

Galvo Mirrors A Galvo mirror (GMs) [] is basically a Galvanometer in principle, except that instead of moving a pointer in response to current, it moves a small mirror. GMs are conventionally used in various laser scanning applications – both precision industrial as well as infotainment like laser shows. As shown in Figure 4(b), two computer-controlled, motorized mirrors are mounted at right angles direct the 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 $\pm 20°$, 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. Typical steering latency is XXXFillmeXXX. [SD: We should say something about MEMS. The reason is that this is something that an optical net person (typical reviewer) will think first as this is used in optical switches. I think there was some text in the earlier version. We can put that back in.]

Tradeoffs Use of SM vs. GM present several design tradeoffs. First, use of GM may obviate the need for additional alignment (e.g., piezo electric positioners). Second, because of the continuous angle, it can reach any receiver within the cone, while SMs provide a small, discrete number of possibilities. But the limitation of GM is that the limited steering angle that makes the network topology dependent on layout geometry and rack locations. Also commodity GM assemblies are much larger than we desire due to the size of the motor as well as the driving electronics. One way to reduce the size is to hide the motor and associated electronics underneath and use custom designed extension arms to hold the mirrors. But additional stability and precision issues must be addressed. This way only optical components will be present on the top of the rack. In general, because the use-cases we envision are significantly beyond the intended applications of either SM and GM, we will systematically investigate the impact of these tradeoffs and we will likely use a hybrid architecture as discussed in Section 4. [SD: check and make sure we discuss.]



(a) FSO prototype on an optical bench set up



(b) Throughput stability

Figure 5: (a) Experimental prototype showing FSO communication using SFP over 7.5m. Note use of mirrors to achieve a long beam path on a standard size optical bench. (The beam is hand drawn.) (b) Distribution of per-second TCP throughputs (in Mbps) over a continuous 30 hour period over 7.5m.

Preconfiguration and Overall Cost Issues Both SMs and GMs will need to be orientated right so that they can form the desired set of FSO links. Since this is to be done in a semi-offline fashion, speed is not of the essence. Also, the actual orientations can be pre-calculated at the installation time. Orientation can be achieved by servos that simply orient the attached component (e.g., SM, GM) to a pre-determined position. Final corrections are to be done using the piezo electric positioners. While we will develop topology design issues for Preconfiguration in Section 4, we consider the actual engineering of this outside the scope of this project. [SD: We told you how to do this at a high level; don't ask further questions.]

Finally, a word about the cost. As is apparent, our goal is to *repurpose commercially available* components to build the flexible FSO links for Firefly. This is to demonstrate feasibility using current generation technologies. However, since the components we will in our actual prototype are built for different use cases and custom work must be performed, the actual cost of our prototype (Section 7) will exceed expectations by several factors. [SD: so that the reviewers do not complain about a link costing $6K.] But our back of the envelop calculations show that with a mass-market design optimized for the DC, the cost an FSO transceiver including the steering support will not exceed about XXXFillmeXXX. [SD: cost is punted.]

3.4 Early Demonstration of Feasibility

We developed a proof-of-concept prototype to demonstrate free space communication using commodity SFPs shown in Figure 5. The prototype uses a pair of 1Gbps SFPs using 1310nm lasers. We launch the beam from a single mode optical fiber 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. We used an achromatic doublet lens 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 to a distance of 7.5m where an identical lens re-focuses beam on the RX detector.[6]

We connect two laptops to the SFPs via standard media converters [] and run TCP throughput experiments for 30 hours to test link stability. Figure 5 demonstrates very stable link performance comparable to the wired case. We also analyzed the sensitivity to misalignment between the TX-RX and found that the throughput is stable up to a transverse shift of ± 0.7 (not shown).

We also independently evaluated the viability of switchable mirrors [**?**] using a 12" x 15" switchable mirror (SM) from Kentoptronics [**?**] tuned for the IR spectrum. 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 confirmed that the FSO beam

[6]Since the SFP used here uses two separate optical paths (for duplex operation), the return link is closed using a regular fiber.

can be reflected from conventional mirrors with no loss in TCP throughput even after multiple reflections.

4 Pre-Configured Flexible Topology Design

The Firefly hardware, 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. Our goal then is to design the most cost-effective and efficient flexible network design that can work within these constraints.

In our context, there are essentially two stages of network design done at different timescales of operation. First, we need to *pre-configure* the network and FSO assembly; e.g., choosing number of FSOs per rack, the specific alignments of the different SMs, or scoping the beam angle of the GMs. This needs to be done at coarse time granularity (e.g., monthly), because of the time incurred in changing such a pre-configuration setup. Second, given this pre-configured setup, we need to choose a *run-time* topology by activating a subset of links, at finer timescales (i.e., few millseconds) based on the prevailing traffic load. Thus, we envision the design workflow in Figure 6. In this section, we focus on the first network design problem done at a coarser timescale, viz., the pre-configuration problem. We defer the runtime operation (called *reconfiguration*) to Section 5.2.

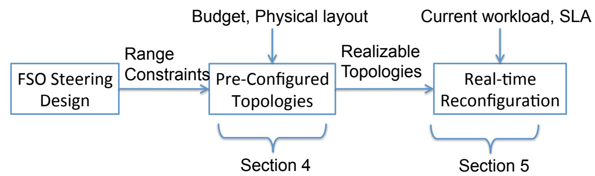


Figure 6: Interaction between the design constraints induced by FSO steering choices, the selection of preconfigured topologies, and the real-time topology selection

Essentially, the pre-configuration problem is: Given an overall budget and phyiscal constraints, 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—that can deliver good performance. This problem is significantly different from prior theoretical work in network design [] on two fronts: (1) *flexibility* requires us to rethink topology design algorithms and traditional performance metrics, and (2) the Firefly hardware elements impose unique physical, budget, and geometric constraints. Thus, we break down our proposed research into two stages. First, to understand the theoretical implications of topology flexibility, we focus on a more abstract problem of designing an optimal "pre-configured flexible topology" (*PCFT*). Then in Section 4.2, we use these insights to revisit the budget-based Firefly network design 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 that each FSO can be steered to, the *pre-configured flexible topology (*PCFT*) design problem* is to determine the set of links connecting pairs of FSOs, so as to optimize the "dynamic bisection bandwidth" of the network (defined below).

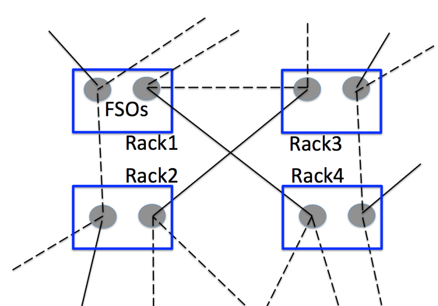


Figure 7: A pre-configured flexible topology (PCTF) with candidate links (solid and dashed). Number of FSOs per rack (m) is 2, and the number of candidate links per FSO (k) is 3. The set of solid (active) represents one possible realizable topology. .

Terminology. A *PCFT* essentially is a k-regular graph over the m FSOs, where each edge is called a *candidate link* and represents an achievable communication link. However, at any point during runtime, only one candidate link per FSO can be *active*; thus, the set of active links form a matching over the FSOs. Given a *PCFT*, any matching over the FSOs is called a *realizable topology* of the given

PCFT. See Figure 7. Thus, the PCFT problem can be thought of as constructing a k-regular graph over nm-nodes such that the *set* of matchings (i.e., realizable topologies) of the graph maximizes the dynamic bisection bandwidth (defined next).

Task 3: *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 a (realizable or static) topology t for the partition p (i.e., the cut-size in t corresponding to p). Then, the traditional notion of bisection bandwidth for a (static) topology t is given by $\min_{p\in P} \mathrm{BW}(t,p)$., while the *dynamic bisection bandwidth* DBW(Π) for the pre-configured topology Π is defined as:

$$\min_{p\in P} \max_{t\in T} \mathrm{BW}(t,p).$$

In addition to just maximizing DBW, we also consider the objective of maximizing DBW under the constraint of bounded network diameter. Network diameter is a reasonable way to bound worst-case latency, which has also been considered as an objective in designing datacenter topologies [**?**]. We also also use an appropriate notion of *dynamic diameter* (as for DBW above) instead of the traditional notion of network diameter.

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

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 closest known problems to this $k = 1$ case of our PCFT problem are: (a) Computing the bisection bandwidth of a given regular graph; this problem is known to be NP-hard [**?**], with the best known approximation-factor of $O((\log n)^2)$ [**?**]. Our PCFT problem is very different from this problem. (b) Determining an upper-bound on the bisection bandwidth of m-regular graphs of size n, for a given m and n; this problem has been addressed extensively, and non-tight upper-bounds (have been determined for small values (upto 4) of m [**?**]. Note that this upper-bound problem can actually be reduced to the $k = 1$ version of our PCFT problem. *T*he above results suggest that the PCFT problem is very likely intractable, even for $k = 1$.

Proposed Approaches: We will pursue three different approaches for the PCFT problem. The first approach builds upon designs with high static bisection bandwidth, the second approach builds upon solutions to special cases of the dual-problem, while the third is a heuristic-based approach.

- *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 have a large girth (length of smallest cycle). 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, for uniform link bandwidths, the inter-rack and inter-machine bisection bandwidths are same. 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[7] (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 (SA-PCFT).* Simulated Annealing (SA) heuristics [] have been used with great success for optimization problems. In our context, since the PCFT design is done offline, we can afford the convergence time incurred by an SA approach. To design an SA approach for our PCFT problem, we need three key components: First, we need good "seed" (starting) solutions; 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 [**?**]). Second, we need 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. Lastly, we need an efficient heuristic for computing DBW (PCTF's objective function) 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 SA-PCFT approach can also be generalize to maximize an appropriately defined **traffic-aware DBW objective**, based on coarse statistics available on inter-rack traffic. 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. 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 incoporating more sophisticated traffic models [].

4.2 Budget-Based Optimization

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

[7]By using $7l$ ports on a ToR switch, we can simulate a Fat tree architecture.

($g \leq m$) 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 4: *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:

- *PCFT-Based Algorithm.* We can use any of the PCFT algorithms as a subroute to solve the BBO problem as follows. 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 constrain k, 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 assume to be a constant. 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. 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.
- *Extending SA-PCFT.* We can modify our Simulated Annealing approach for the PCFT problem to solve 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, g, 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 building blocks from previous sections, to implement a feasible reconfigurable datacenter network.

5.1 System Overview

We first describe the high-level roles of the different components of the management layer. See Figure 8.

- **Monitoring Engine (ME):** ME provides network status information to the management layer. E.g., it provides (i) status of individual inter-FSO links, (ii) measurements of observed traffic patterns such as inter-rack *traffic matrix*) or views of "elephant" flows [**?**, **?**, **?**].
- **Optimization Engine (OE):** Given the offered traffic workload, a pre-configured flexible topologies (PCTF), and the current network state (e.g., active links and link status), the optimization engine devises an efficient *reconfiguration and traffic engineering strategy* so as to achieve desired performance goals (e.g., throughput, latency).
- **Data Plane Translation Engine (DPE):** DPE translates the output of the optimization engine into a data plane strategy.
- **Application APIs:** The management layer also provides APIs to the users/tenants to best leverage the benefits of reconfigurability. E.g., users can use APIs to inform relevant application details (e.g., expected traffic patterns, single/multi-path TCP) to the optimization and data plane modules.

Challenges. In designing the Firefly management layer, we build on traditional cloud and network management including traffic engineering [], software-defined networking [], fast routing recovery [], and managing network updates []. The key differences from these prior works arise on

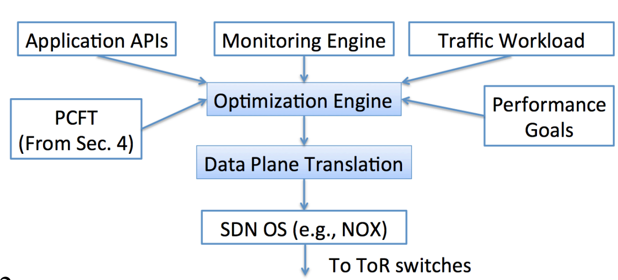


Figure 8: Overview of the Firefly management layer

three dimensions. First, prior traffic engineering efforts typically assumes the network (topology) to be *static*, whereas the Firefly network is inherently *dynamic*. This gives rise to new challenges and opportunities for topology reconfiguration, traffic engineering, and data plan strategies. Second, constraints imposed by the free-space optics (e.g., reconfiguration latency) and pre-configured topologies give rise to unique challenges w.r.t connectivity and performance guarantees during reconfigurations, 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.

Plan. In the following susections, we address OE, DTE, and the control channel implementation, respectively. 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 Reconfiguration and traffic engineering

Task 5: *We will develop fast, near-optimal algorithms for the joint optimization problem of 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.*

Joint Reconfiguration and Traffic Engineering (JRTE) Problem. Given the traffic load, the *JRTE* problem is to: (a) Select a realizable-topology from a given pre-configured flexible topology (recall that a realizable-topology is a matching of candidate links over the FSOs); and (b) Route the given inter-rack flows over the *selected* realizable topology, so as to optimize a desired objective. The second traffic engineering (TE) part essentially involves solving the multi-commodity flow problem over the realizable topology. For the latter, the set of FSOs on each rack are considered as one node (since they are connected by a ToR switch). The objective functions of interests could be to minimize link congestion or maximize total flow in conjunction with fairness, latency bound, and/or tenant provided SLAs.

Connection to Prior Works For the special case when the given PCFT has exactly one realizable-topology (i.e., the given candidate links already form a matching over the FSOs), our JRTE problem is exactly the NP-hard [] multi-commodity flow problem with the desired objective, assuming the given TCP flows to be unsplittable [**?**]. Thus, our JRTE problem is trivially NP-hard. The reconfiguration subproblem can be looked upon as a kind of topology control problem [], which is to establish/select links between given wireless nodes to achievenetwork connectivity while minimizing the transmission power (or energy consumption) of the nodes. The constraints and objectives of the topology control problems are quite different than our reconfiguration subproblem, and thus, the techniques used for topology control are not directly applicable. Finally, the reconfiguration subproblem (as the PCFT problem) also falls in the class of *degree-constrainted subgraph* problem, but the TE-based objectives makes the reconfiguration subproblem very different from prior-addressed degree-constraintted subgraph problems.

Proposed Approaches. We note that the ILP formulation of the JRTE problem took several hours to solve on the state-of-art ILP solver [], even for a small 20-node instance. In contrast, the reconfiguration of our Firefly network should not take more than a few milliseconds, for it to be of any real benefit. Thus, the challenge is to design very fast, scalable, and efficient JRTE algorithms.

- *Matching Techniques.* One simple and reasonable way to approach the reconfiguration subproblem would be to select the maximum-weighted matching (solvable in polynomial time) between FSOs, where the link (i, j) is weighted by the inter-rack traffic demand between the correspondin racks. Such a topology essentially serves the maximum possible total inter-rack demand using *one* hop paths. It is challenging to generalize the above approach to include two-hop routes, i.e., to select the matching that serves the maximum traffic demand in one or two hops. In general, we would like to generalize the approach to include multi-hop routes and define the weight of the matching appropriately (to favor shorter routes). Even generalizing the matching algorithm to ensure that the corresponding inter-rack graph is connected is challenging, but this may be a reasonable tractable objective. Following the above, we can do the TE part independently using standard techniques [] over the selected realizable-topology.

- *LP Relaxation Techniques.* One promising approach is to formulate the reconfiguration problem as an ILP (using flow-like constraints and binary variables for link selection) with the objective of minimum link congestion or maximum "fair" flow, and solve the relaxed LP. We can then convert the LP solution to an ILP solution by an appropriate "rounding" technique, while ensuring that the "matching constraint" is still satisfied (unsatisfied flow-constraints will only result in a sub-optimal TE solution). Here, for the unsplittable (i.e., single-path routing) verion, we also need to do path-striping as in [**?**] in conjunction with the rounding process. The above LP relaxation approach is similar to the following alternate approach: First solve the multi-commodity problem over the entire PCFT graph, and then select a "good" matching based on the flow values on the links. Both approaches are expected to be fast, and it would be interesting to compare their relative performance over real traffic traces.

Further Directions. In addition to the above, we are also interested in designing online or distributed reconfiguration algorithms.

- *Limited Traffic Predictability; Online Reconfiguration.* Our discussion implicity assumes the availability of traffic loads for the next epoch. However, in reality, traffic predictability may be limited. In the worst case, we may only be able to distinguish between "elephant" (large) and "mice" (small) flows, based on their initial size. In such restricted settings, the reasonable approach would be to change the realized topology in an online manner in response to the arriving elephant flows, while relying solely on TE for the "mice" flows. This approach should be effective since the structure of real-world workloads suggests that a small number of "elephant" flows carry the most bytes []. Moreover, since these elephant flows are typically long-lived [], they are quite amenable to coarser time-scale optimizations. In our preliminary work [**?**], we employed a simple strategy along the above directions, and achieved near-optimal performance over randomly generated traffic traces. More information about traffic loads such as spatial and temporal distribution of elephant flows (or flow sizes in general) would require challenging generalizations of the above approach. An addition challenge to address in an online strategy would be to favor reconfiguration that cause minimal disruption to ongoing traffic flows.
- *Estimating Reconfiguration "Impact".* The above approach suggested that the topology of the network need not be changed in response to newly arrive mice/small flows. In general, we would like to be able to estimate the "impact" of reconfiguration (topology change and/or TE), so that we commit to the suggested topology change and/or TE only in effective cases. We could define the *impact* in terms of the expected decrease in evacuation time, latency, and/or number of dropped packets. Computation of such appropriately defined impact may be intractable, but upper and/or lower bounds on its value may be sufficient and very useful for our purposes.
- *Incremental or Localized Reconfiguration.* One of the ways to develop a fast and effective reconfiguration algorithm is to determine the required topology change in an "incremental" manner (e.g., by constraining the number of links that need to deactivated or activated), and moreover, even limiting ourselves to only localized (w.r.t. to the traffic changes) reconfiguration. For the former, we can use augmenting-path techniques to incrementally improve the matching [] and/or change the objective function appropriately. For the latter, we can exploit the "flow" structure of the optimization to design distributed strategies [**?**, **?**]. Note that localized reconfigurations may result in multiple concurrent re-

configurations across the network, which would need to be handled carefully to guarantee consistently (as discussed in the next subsection).

5.3 Implementing data plane strategies

Task 6: *We will design and implement efficient data-plane implementations to guarantee reachability and consistency properties, in presence of reconfiguration and link dynamics.*

In translating the solution provided by the optimization engine into a consistent and efficient data plane forwarding strategy, We build upon recent advances in software-defined networking (SDN). While SDN is an "enabler," as it provides cleaner management abstractions and open interfaces (e.g., via APIs such as OpenFlow []), Firefly introduces unique consistency and efficiency challenges. In particular, in face of a dynamically changing network due to reconfigurations, we need to ensure that (a) packets do not use deactivated links (i.e., black holes [] are avoided), (b) the network remains connected at all times, and (c) the packet latency remains bounded. Finally, we also need a data plane strategy to: (d) handle transient misalignment of FSO links. We note that the recent related works either assume a *static* network [**?**, **?**] or focus on a single reconfiguration [**?**], and hence, are not directly applicable to our context.

(a) and (b). Guaranteeing Correctness and Connectivity. Packets are routed in the network on the basis of forwarding tables, which essentially specify, at each node, the next hop/link to use for each destination. In a dynamic network, forwarding tables will also be changing constantly. Note that activation/deactivation of a link takes a finite amount of time, and that we cannot update the tables across all the network switches atomically (i.e., *at once*). In face of the above challenges, we need to ensure that through every possible intermediate state of the links and switches' tables, only active links appear in the forwarding tables. We can ensure this by a careful ordering of steps as suggested in our preliminary work [**?**]; in particular, (i) we reflect removal of links in the forwarding tables, before actually deactivating the links, and (ii) reflect addition of links in the tables only after the link activation is complete. Note that the above solution ensures the desired property even in face of multiple concurrent reconfigurations, and irrespective of the order in which forwarding tables are updated across the network.

In addition to above, we also need to ensure that the network remains connected at all times. There are two possible options: (i) We maintain a static "backbone" subnetwork that ensures connectivity, or (ii) reject reconfigurations that disconnect the network. The first approach reduces the degree of flexibility in network design and may result in high packet latency, depending on the backbone. The second approach becomes challenging to implement if there are multiple *concurrent* reconfigurations. There are three options to handle concurrent reconfigurations: (i) one at a time, (ii) in batches (i.e., queue and combine them into a single reconfiguration); and (iii) execute each reconfiguration individually but *concurrently*. The first two options can be inefficient as large flows have to wait until the desired link(s) become available, while the third option requires a careful implementation to ensure consistency. In particular, for the third option, we need to keep a single consistent view of the network topology graph and allow only *atomic* access to it (when one needs to check if deactivation of a set of links disconnents the network). Again, the above solutions work irrespecitve of the order in which the forwarding tables are updated across the network. In our research, we will study the performance of the above described approaches.

(c) Guaranteeing Bounded Packet Latency. The above strategies still do not guarantee a bounded packet latency. In fact, in general, its *impossible* to avoid guarantee bounded packet latency in general. See Figure 9. There are two approaches to bound the latency of (in-flight) packets: (i) create and use a backbone *static* subnetwork with bounded diameter, (ii) reject reconfigurations to avoid high packet latency. The first approach will require careful decision-making of *when* to resort to routing a packet to the backbone (due ot its limited bisection bandwidth), while the second

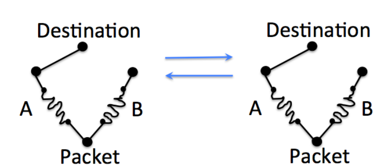


Figure 9: If the network goes back and forth between the above topologies (due to corresponding reconfigurations), then the packet will continue to "swing" between areas A and B – leading to a forwarding "loop".

approach will require an efficient and fast computation of the impact on latency of the packets (especially, the in-flight packets). In addition, we should formally characterize and avoid scenarios like the one described in Figure 9.

(d) Handling Misalignment of Links. In Firefly, even during a static topology state, links may be temporarily unavailable because of possible misalignment of the FSO links. Such misalignments are fixed in real-time by "micro alignment" of FSO devices, as suggested in Section 3, and the timescales of such micro-alignment is likely to be much smaller than the time needed to update rules [**?**] through an SDN controller. In fact, it may even be counterproductive to report such transient link failures to the controller, as it may cause needless reconfigurations and/or update of forwarding tables. Thus, we need appropriate network layer techniques to recover from such transient link failures. 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 in our research. In the absence of such features, we will investigate design of a local "lightweight" SDN controller on every rack that can quickly react to such misalignments while relying on the global controller for longer-timescale reconfigurations [**?**].

5.4 A wire-free control channel

Task 7: *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.[8]

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 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 de-

[8]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.

scribed 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.

[VS: feels a bit adhoc .. can we have some story – do these extensions tackle new opportuntieis or performance improvements?]

- *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 inter-connections between these switches, and the PCFT solutions would involve more general non-regular graph. [VS: would drop this unless we see strong result otherwise from the simulation.. it seems to goes against the wireless/FSO vision? or are you thinking FSO to these as well?] HG: Yes, I am thinking of using FSO for the steiner switches, hence the paragraph heading.
- *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 8: *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:** We will prototype a proof-of-concept 10 Gbps SFP-based FSO devices with a small form factor and design optical mechanisms to collimate the laser beam to about 100m. prototype two proposed steering mechanisms: using switchable mirrors and galvo motors. As a starting point, we will decouple these two steps and repurpose our existing commodity/outdoor FSO devices [] to test steering mechanisms.
- **Reliability of steerable FSO in realistic conditions:** Real DCs will have several sources of "disturbances" (e.g., rack vibration, temperature gradients, airflow patterns, etc.) that may cause alignment and performance issues for FSO communication. First, we will create a lab environment that can emulate the effects of different types of disturbances. To estimate the range parameters for these effects, we will engage our industry partners (see letters from Facebook and Microsoft) and add 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 in an actual DC environment (CEWIT cluster in Stony Brook University) and conduct a longitudinal study of the reliability of the links.

  [SD: do we have access to a lab that can emulate "disturbances"??]
- **Performance and benefits under realistic workloads:** We will develop scalable packet- and flow-level simulation platforms extending prior work [**?**, **?**] to evaluate the benefits of our topology design (Section 4) and reconfiguration (Section 5) algorithms. We will start with extrapolating from existimg small-scale datasets [**?**, **?**, **?**, **?**] and work with industry supporters (e.g., Facebook and Microsoft) to quantify the benefits at scale.
- **Responsiveness, and correctness of control plane:** We will implement a SDN controller starting with research prototypes [**?**] and port our ideas to open-source platforms such as OpenDayLight [] as the project matures. We will synthesize benchmark suites to "stress-test" the scalability and responsiveness of our controller. We plan to leverage our experiences with emulation platforms such as MiniNet [] and Emulab [] to test the correctness of the proposed recovery and consistent reconfiguration mechanisms in the presence of network dynamics.
- **End-to-end integration and evaluation:** A full-scale DC testbed is outside the scope of the proposal in terms of infrastructure and personnel resources.[9] Within the scope of our budget, we will 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 enable precise timing and diagnostic information [], link characterization [], as well as aid in high-rate traffic generation [].

  The 4 node setup (along with the 4x3=12 FSO devices) will be deployed on top of the racks in an operational cluster (in CEWIT). 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?]

  [VS: something abt USRP etc?]

8 Broader Impact

Some input from Jon would be helpful too.

Impact on Economy and Environment. With growing interest in Big Data, cloud computing and virtualization, data centers are now common in every sector of the economy. This includes IT industry, government, media, healthcare, financial sector, transportation and the scientific community. The largest of the data centers are known to cost more than a billion USD and are significant power hogs consuming 10s of MW of power []. Overall, recent EPA studies concluded the the total data center electrical power usage is roughly

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

a few percent of the entire electricity consumption in the US and lagging only modestly behind the total household electricity consumption []. We foresee that the Firefly architecture can significantly reduce both cost (by eliminating the need for over-provisioning) and energy consumption (by making the network design energy-proportional and also by improving cooling). This certainly will have perceptible economic impact by making many IT services cost less - both in terms of dollars and carbon footprint - across all sectors in the economy. In addition, success in the proposed project will garner immediate interest in industry for further developing and productizing the proposed FSO-based interconnection. R&D and manufacturing of such interconnections will produce a different form of device industry that will include optical engineers in addition to traditional computer hardware engineers.

Integration of Research and Education. A strength of the project is that it brings together two disparate disciplines, opto-electronics and computer systems. The project will directly contribute to graduate courses in both mechanical engineering and computer science, especially by having relevant project topics and lab support available to the students. We also plan to develop tutorial materials on data center networking and FSO communications, present such tutorials in relevant conferences and finally make them available freely via YouTube.

[VS: probably need some concrete pointers here on wireless classes, SDN/advanced classes, theory classes etc that we have taught and generated some tangible research from]

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. SBU has a Simons Summer Research Program[10] that provides a mechanism to recruit talented high school students. The PIs have contributed to in the past. Students in the this program routinely competes nationally in the Intel Science Talent Search and often successfully with SBU professors as mentors. The PIs will also use REU supplements to engage undergrads in their 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 students. 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 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.

[VS: is there concrete evidence .. this seems to say someone else in SB has done this, not necessarily us :)]

[VS: are the kits budgeted for?]

Involving Under-Represented Groups. SBU has a history of active outreach efforts in order to involve traditionally under-represented groups in science and engineering research. This includes the Turner Fellowship Program minority for graduate students, 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.

[VS: probably should say something abt pis track record in working with underrepresented gro?]

[10] 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.