ProjecToR: Agile Reconfigurable Data Center Interconnect

Monia Ghobadi Ratul Mahajan Amar Phanishayee
Nikhil Devanur Janardhan Kulkarni Gireeja Ranade
Pierre-Alexandre Blanche[†] Houman Rastegarfar[†] Madeleine Glick[†] Daniel Kilper[†]
Microsoft Research [†]University of Arizona

Abstract— We explore a novel, free-space optics based approach for building data center interconnects. It uses a digital micromirror device (DMD) and mirror assembly combination as a transmitter and a photodetector on top of the rack as a receiver (Figure 1). Our approach enables all pairs of racks to establish direct links, and we can reconfigure such links (i.e., connect different rack pairs) within 12 μs. To carry traffic from a source to a destination rack, transmitters and receivers in our interconnect can be dynamically linked in millions of ways. We develop topology construction and routing methods to exploit this flexibility, including a flow scheduling algorithm that is a constant factor approximation to the offline optimal solution. Experiments with a small prototype point to the feasibility of our approach. Simulations using realistic data center workloads show that, compared to the conventional folded-Clos interconnect, our approach can improve mean flow completion time by 30–95% and reduce cost by 25–40%.

CCS Concepts

•**Networks** → **Network architectures;**

Keywords

Data Centers; Free-Space Optics; Reconfigurablility

1. INTRODUCTION

The traditional way of designing data center (DC) networks—electrical packet switches arranged in a multi-tier topology—has a fundamental shortcoming. The designers must decide in advance how much capacity to provision between top-of-rack (ToR) switches. Depending on the provisioned capacity, the interconnect is either expensive (e.g., with full-bisection bandwidth) or it limits application performance when demand between two ToRs exceeds capacity.

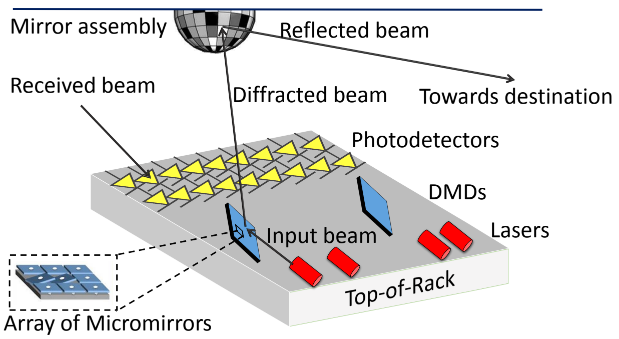


Figure 1: ProjecToR interconnect with unbundled transmit (lasers) and receive (photodetectors) elements.

	Enabler Tech.	Seamless	Fan-out	Reconfig. time
Helios, c-Thru, Proteus, Solstice [16, 26, 37, 38]	OCS	No	100-320	30 *ms*
Flyways, 3DBeam [23, 40]	60GHz	No	≈70	10 *ms*
Mordia [33]	OCS	No	24	11 μs
Firefly [22]	FSO	Yes	10	20 *ms*
ProjecToR	FSO	Yes	18,432	12 μs

Table 1: Properties of reconfigurable interconnects.

Many researchers have recognized this shortcoming and proposed reconfigurable interconnects, using technologies that are able to dynamically change capacity between pairs of ToRs. The technologies that they have explored include optical circuit switches (OCS) [16,25,26,33,37,38], 60 GHz wireless [23,40], and free-space optics (FSO) [22].

However, our analysis of traffic from four diverse production clusters shows that current approaches lack at least two of three desirable properties for reconfigurable interconnects: 1) *Seamlessness:* few limits on how much network capacity can be dynamically added between ToRs; 2) *High fan-out:* direct communication from a rack to many others; and 3) *Agility:* low reconfiguration time.

Table 1 compares the existing reconfigurable interconnects with respect to these three properties. Most approaches (rows 1–3) are not seamless because they use a second, re-

SIGCOMM '16, August 22-26, 2016, Florianopolis , Brazil

DOI: http://dx.doi.org/10.1145/2934872.2934911

configurable technology on top of the electrically-switched network. This design places a hard limit on the amount of network capacity that can be dynamically reconfigured. Some OCS-based interconnects (row 1) have relatively high fan-out, but still not enough to allow a ToR to reach every other ToR for large DCs (e.g., 1000 racks). FireFly [22] is seamless but has low fan-out and low agility.

We propose a new way to build reconfigurable interconnects. Our proposal, called ProjecToR, uses FSO between racks as the basis for all traffic. Its high fan-out and high agility is enabled by digital micromirror devicees (DMDs), which are ubiquitous in digital projection technology. DMDs can steer light in tens of thousands of directions, depending on their configuration, and they can switch between different directions in 12 μs. Prior work has used DMDs for low port-count (7–32) optical switches [28, 32]; we explore their use in building a DC-wide interconnect.

An immediate challenge for our exploration is that DMDs have limited angular range of $\pm 3^\circ$, and all possible directions of light lie within this range. This low range limits the physical space that can be covered by the DMD, nullifying its fan-out advantage. We overcome this limitation by pointing the DMDs toward a "disco-ball" mirror assembly installed overhead. The assembly's angled facets magnify the DMD's reach to the entire DC.

Figure 1 illustrates our design. Instead of using conventional transceivers, which bundle a laser (to transmit light) and a photodetector (to receive light), we unbundle these components. A laser shines light on the DMD, steering it toward a facet on the mirror assembly, which then reflects it toward the receiver. Exposed photodetectors on destination racks act as receivers, helping retain the reconfiguration speed offered by DMDs because the receivers do not need to be reconfigured in any way based on the sender.

In this paper, we build a small, three-ToR prototype of ProjecToR and develop and evaluate algorithms to route traffic in this type of interconnect. We leave for future work other important questions such as the impact of dust and vibration on the stability of FSO links.

The traffic routing challenge in ProjecToR stems from the fact that its optical setup is a "sea" of transmitters and receivers that can be linked in a multitude of ways. We divide possible links into two categories: *dedicated* and *opportunistic*. Dedicated links carry small flows, possibly over multiple hops, and change configuration at coarse time scales (e.g., daily). Opportunistic links carry large flows over single hops and change rapidly based on current demand. Our two-topology split is conceptually similar to earlier two-technology approaches but a fundamental distinction is that, being based on the same technology, we change how resources are split across ToRs and across time.

The problem of scheduling opportunistic links is akin to that in switch scheduling [29, 30], but with an important distinction: our setting is "two-tiered." While the traffic demand is between ToRs, links are between lasers and photodetectors, and many laser-photodetector combinations can serve traffic between a pair of ToRs. Current switch scheduling algorithms do not tackle this two-tier case. We develop an algorithm based on stable matching [17] that is provably within a constant factor of an optimal oracle that can predict traffic demands. It can be implemented in a fully decentralized manner, thereby allowing it to scale to large DCs. Our scheduling algorithm may be of independent interest because two-tier scheduling can arise elsewhere (e.g., if a ToR has multiple links to a high port-count optical switch).

By conducting experiments on our prototype, we show that DMD-based FSO communication can provide throughput comparable to optical fiber cables, can cover long distances, and can switch rapidly between receivers. Our large-scale simulations show that, compared to a full-bisection, electrically-switched network and FireFly, ProjecToR can improve flow completion times by 30-95%. Based on component costs, we estimate a ProjecToR interconnect will be 25-40% cheaper than a full-bisection network.

2. MOTIVATION

We use traffic traces from over 200K servers across four production clusters to motivate reconfigurable interconnects and identify their desirable properties. We are not claiming that these clusters are similar to all others but they do represent diverse systems in production today.

Our clusters run a mix of workloads, including MapReduce-type jobs, index builders, and database and storage systems. They have between 100 and 2500 racks, and we name them *Cluster1* through *Cluster4* based on their size.

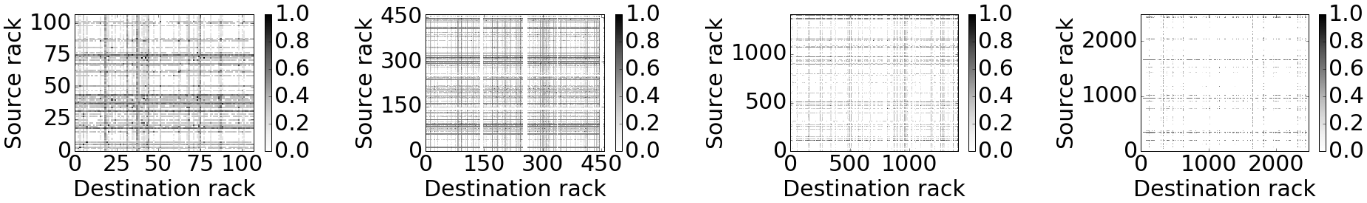
We instrument each server to log application demand, by recording the number of bytes written as part of socket calls. We aggregate these logs into a series of rack-to-rack traffic matrices, where each matrix represents the demand between pairs of racks in 5-minute, daily and weekly windows.

2.1 Need for seamless reconfigurability

Figure 2 shows a heatmap of rack-to-rack traffic, for representative five-minute windows for each cluster. Rows correspond to source racks and columns to destination racks, while the color encodes the amount of traffic from the source to the destination. Thus, horizontal lines are for racks sending a lot of traffic to many other racks and vertical lines are for racks receiving a lot of traffic from other racks. The colors are normalized to the maximum rack-to-rack traffic and the scale is logarithmic (i.e., 1.0 corresponds to log_{10} of the maximum traffic). Without logarithmic scaling, almost the entire heatmap appears white, with a few dots; the traffic is skewed and the largest pairs dominate.

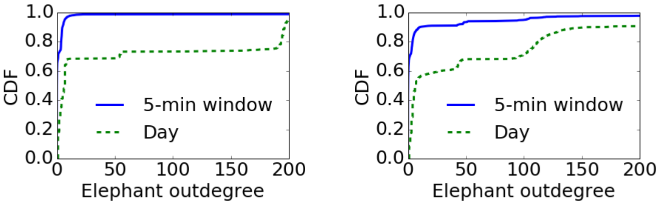
Despite such scaling, we see that much of the heatmap is white for each cluster, that is, many rack pairs exchange little or no traffic. Data reveal that 46-99% of the rack pairs exchange no traffic at all, while only 0.04-0.3% of them account for 80% of the total traffic. Topologies that provide uniform capacity between every pair of racks (e.g., Clos fabrics [10, 21]) are, thus, either over-provisioned with respect to most rack pairs or under-provisioned with respect to the rack pairs that generate a lot of traffic.

Such observations have led researchers to argue for reconfigurable topologies that can dynamically provide addi-



(a) *Cluster1* (b) *Cluster2* (c) *Cluster3* (d) *Cluster4*

Figure 2: Heatmap of rack to rack traffic. Color intensity is log-scale and normalized by the maximum traffic between any two racks.



(a) *Cluster3* (b) *Cluster4*

Figure 3: Elephant outdegree of source racks.

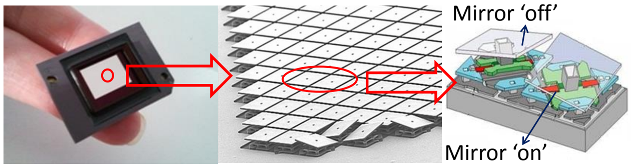


Figure 4: A DMD chip (left), micromirrors close up (center), and two side-by-side micromirrors showing the two possible mirror positions (right) [12,34].

tional capacity between hot rack pairs. Most designs, however, provide reconfigurability by augmenting the uniform-capacity electrical packet switching (EPS) network with another technology (e.g., optical switching [16, 38] or RF [23, 40]) that provides non-uniform capacity between rack pairs. The parallel network carries some or all of the large flows, while the EPS network carries the remaining traffic, including short flows. But given the differences across our clusters, it is difficult to determine in advance the relative capacities needed for the two networks. This approach has another downside, in that the hardware for the parallel network needs provisioning for all racks, because, in theory, any rack can source or sink a large flow. This need leads to inefficiency and higher cost. For instance, an optical switch with a large number of ports is needed to connect all racks, even though only a small fraction of pairs will exchange large flows.

We thus argue for seamless reconfigurability based on a single technology. The topology can adapt to any traffic matrix and use all available resources to provide *i*) high capacity between the rack pairs that currently need it, and *ii*) low delays for the remaining traffic. Below, we further argue that reconfigurable topologies need high fan-out and agility.

2.2 Need for high fan-out and agility

The fan-out of a reconfigurable topology is the number of direct links that a rack can create to other racks under different configurations (not simultaneously). High fan-out is valuable if the traffic is such that source racks send large amounts of traffic to many other racks. Without it, large flows will traverse multiple hops and consume the capacity of multiple links.

In our clusters, source racks often do indeed send large traffic quantities to many other racks. This behavior can be seen as dark horizontal lines in Figure 2 and more directly in Figure 3. We define the *elephant outdegree* of a source rack as the number of destination racks to which it sends an *elephant* transmission. For this analysis, the set of elephant transmissions are arbitrarily defined as those that collectively carry 80% of the traffic; our essential observation does not change if we raise or lower this threshold. Figure 3 shows the elephant outdegrees for the two largest clusters. We see that the elephant outdegree of 15-25% of the source racks is more than 150 over the course of a day. Even within a five-minute window, some source racks send elephants to over 100 unique destinations. Roy et al. made a similar observation for traffic matrices in Facebook's data centers [35].

High elephant outdegree motivates not only high fan-out but also agility, i.e., low switching time. If a source rack is sending elephants to many destinations, it must be able to switch quickly among those destinations. Otherwise, applications for destinations that are made to wait long will suffer.

Unfortunately, as noted earlier, none of the reconfigurable solutions today meet even two of three requirements above and some lack all three. These observations motivated us to explore a different approach in ProjecToR.

3. BACKGROUND ON DMD

Our approach is based on DMDs. Before outlining it in the next section, we provide a brief background on these devices. DMDs are at the core of digital projection technology today [8]. As shown in Figure 4, they are two-dimensional arrays of micromirrors, where each micromirror can be switched between on/off positions [12]. They can be used as diffractive optical elements to direct light through free-space. By changing micromirrors' on/off positions, the direction of the diffracted light can be finely tuned. DMDs are available in multiple resolutions such as

768×1024, which reflects the number of rows and columns of micromirrors, respectively.

To use a DMD as a switch, it is configured with a computer generated pattern (CGP). The CGP is a 0/1 image with resolution equal to that of the DMD. A '0' ('1') at pixel (x,y) means the micromirror in that pixel is in the off (on) position. The on/off arrangement of all micromirrors together creates a diffraction effect that determines the direction of the light. Given a desired direction for the diffracted light beam, we can calculate the CGP using Gerchberg-Saxton iterative Fourier transform algorithm (IFTA) [19]. No manual manipulation is required to switch light and DMDs can support switching speed as low as 12 μs (§6).

By controlling the on/off positions of mircromirrors, a DMD can steer light towards thousands of directions, enabling a high fan-out. Typically, an N×N DMD can address $\frac{N\times N}{4}$ independent points with negligible crosstalk between adjacent points [18]. In projection systems, the ratio of 4:1 is commonly used. In communication systems, sensitivity to crosstalk reduces it to 32:1. Thus, for the 768×1024 DMD used in our prototype, we can reach approximately $\frac{768\times768}{32}$ = 18,432 points. We generate a cylindrical beam and approximate that its circular cross-section covers 768×768 micromirrors. Higher resolution DMDs, which are commercially available [9], will yield an even higher fan-out.

Given this high fan-out, we assume in this paper that every ToR can directly reach every other ToR. That is, there is at least one laser on a source ToR that can reach at least one photodetector on the destination ToR. This constraint can be easily met for even the largest DCs. For instance, a DC with 100K servers and 50 servers per rack will have 2K ToRs. These numbers imply that each source laser should be able to connect on average to over 9 (≈18,432 / 2K) photodetectors per destination ToR.

While the focus of this paper is on a free-space interconnect, the properties of a DMD make it suitable for an ultra-high port-count optical switch. Many of our techniques, e.g., the mirror assembly and scheduling algorithms, can be packaged within a switch as well. In contrast, current Micro-Electro-Mechanical Systems (MEMS)-based optical switches have a port count in 100s, and there is a fundamental limit to their scaling [31].

4. OVERVIEW

While DMDs have many attractive properties, building a DC-wide interconnect based on them is not without its challenges. One challenge is to engineer reach: light coming out of the DMD should be able to reach receivers all around it, some of which are far away. Because DMDs have a narrow angular range of ±3°, all 18,432 unique angles occur within this range. Thus, in any given orientation, the DMD will be able to reach only a small subset of the receivers, negating the benefits of its theoretically high fan-out.

We address this challenge by coupling DMDs with a mirror assembly that has many facets oriented at different angles. There is one facet per intended receiver; its angle depends on the relative orientation of the DMD and the receiver. Given the locations of all the receivers a DMD is intended to reach, we can design the mirror assembly in a way that accounts for the DC geometry and alignment tolerance. We outline a possible design in a separate report [20].

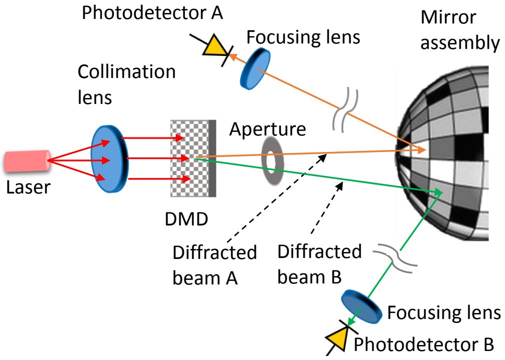


Figure 5: Detailed optical setup.

Figure 5 shows this optical setup in detail. The laser shines modulated light (i.e., carrying digital information) through a collimation lens, which transforms the light into a cylindrical beam. The focal length and diameter of the lens is selected such that the beam's diameter matches the size of the DMD. When this beam falls on the DMD, it is diffracted in a direction that is based on the CGP loaded into the DMD. After being diffracted by the DMD, the beam enters an aperture that selects a diffraction order able to hit an angled facet on the mirror assembly. The facets have fixed orientations that reflect the light toward a specific destination. A lens focuses the beam onto a photodetector at the destination. In our current design, each laser has its own DMD, and each DMD has its own mirror assembly. In the future, we will explore sharing DMDs across lasers [28] and sharing mirror assemblies across DMDs.

The combination of diffraction by the DMD and reflection by the mirror assembly enables the laser to access a large number of photodetectors on top of different racks. To transmit to the desired photodetector, we load the appropriate CGP into the DMD. We calculate all CGPs in advance and store them in a lookup table at the ToR. No computational delay occurs during switching.

A remarkable advantage of our optical setup is that it is completely modulation agnostic; it can scale to higher bandwidths without the need to change anything in the interconnect except the transceivers because only the circuitry behind the lasers and photodetectors is modulation-specific. DMDs and mirror assemblies simply steer light. Thanks to this property we can (selectively) upgrade source and destination modulation, without touching the rest of the interconnect. In traditional wired topologies, changing source-destination modulations requires changes to intermediate switches as they modulate and demodulate light.

As an aside, while the data network in ProjecToR is fully wireless, we retain a wired *management* network that connects ToRs via their management port.

To be feasible, our design must address several chal-

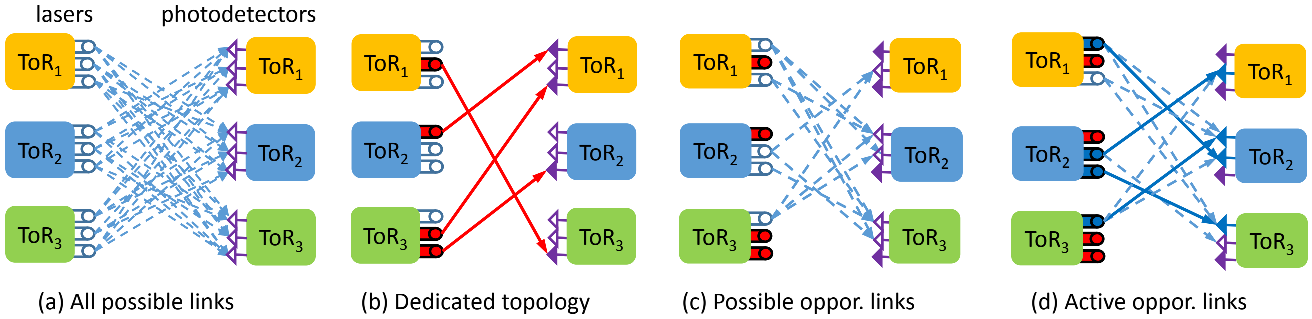


Figure 6: Splitting lasers and photodetectors into dedicated and opportunistic between three ToRs. This example assumes every laser can connect to two photodetectors on other ToRs. The dedicated topology is built based on average demand and changes on a slow time scale. The opportunistic topology is built based on current demand and changes on a fast time scale.

lenges, some of which are related to the physical properties of data centers. For instance, we must ensure the robustness of free-space links to any vibration of the racks and to any dust on optical components. We must also ensure appropriate clearance above the racks to mount the mirror assemblies and ensure that the light path is not blocked by any infrastructure component. We are exploring these challenges in ongoing work.

In this paper, we focus on a different but equally-important challenge: *how should packets be routed over this incredibly flexible interconnect?* Lasers and photodetectors can be connected for transmissions in many ways—each source ToR can select from multiple lasers to send packets to one of many photodetectors on the destination ToR. Of course, there are desirability and feasibility constraints. For example, two lasers should never point to the same photodetector. In addition, while we can adapt to traffic demand, there is a cost to changing the combination—it takes the DMD 12 μs to load the appropriate CGP and establish connectivity to the intended receiver. As we go on to explain in the next section, we handle this packet routing challenge by using a novel approach that operates at two time scales. We detail our approach next.

5. BUILDING TOPOLOGIES

The ProjecToR fabric is a flexible interconnect in which lasers and photodetectors can be connected in many ways, each resulting in a different topology. One extreme approach to making these connections is to have a fast-changing topology: connect lasers and photodetectors based entirely on the *expected* traffic matrix in the near future. This approach can provide good connectivity if future traffic is predictable, but accurately predicting traffic is difficult, and flows whose arrivals cannot be predicted in advance may suffer. Such flows may not have good paths between their endpoints when they arrive and must wait for good paths to be created through topology reconfiguration. Reconfiguration delays are especially hurtful for short flows.

The other extreme is to build a slowly-changing topology among lasers and photodetectors, based on the *average* traffic matrix across time (rather than what is expected in the near future). While this approach can provide good paths for most flows, independent of their arrival, it can be extremely sub-optimal for some flows. This sub-optimality will cause high collateral damage when such flows are large, as they will consume capacity across multiple hops in the topology. Large flows should ideally be carried over direct paths.

In ProjecToR, we strive to get the best of both worlds. We use a subset of lasers and photodetectors to form a multi-hop *dedicated* topology. This topology changes on a slow time scale (e.g., daily) and serves as the default for all flows. The remaining lasers and photodetectors establish single-hop *opportunistic* links on a fast time scale based on demand from the large flows under-served by the dedicated topology. Figure 6 illustrates our two-topology approach for a case with three ToRs each with three lasers and photodetectors. For clarity, the sender side is separated from the receiver side.

Our approach is reminiscent of reconfigurable topologies such as Helios and c-Through [16, 38]; the dedicated topology is akin to the electrical network and the opportunistic topology maps to the optical network. However, we have the ability to dynamically alter the amount of resources allocated to each topology and to allocate different amounts of resources per ToR to each topology (e.g., more opportunistic resources at heavy-sending ToRs). These capabilities enable good performance for a broad range of traffic matrices.

With our overall strategy in place, we have four tasks: *i*) allocating resources among dedicated and opportunistic topologies; *ii*) connecting dedicated resources into a network; *iii*) routing traffic over this topology and moving under-served flows to opportunistic links; and *iv*) connecting opportunistic resources and transferring data. The first two tasks are infrequent (e.g., daily) and based on historical traffic; the last two are on the order of micro seconds.

All flows start on the dedicated topology, which uses k-shortest path routing. If a flow accumulates a *bundle* of packets at the source, it is classified for opportunistic transmissions and waits for the opportunistic scheduler to serve it. A bundle is a unit of transmission on opportunistic links, and all its packets have the same source and destination ToRs. The number of packets in a bundle is a trade-off between system efficiency and latency. Small bundles hurt efficiency because of the cost of reconfiguring opportunistic links (12 μs in our case). Large bundles can hurt latency if head-of-line blocking occurs, and other bundles destined to different ToRs have to wait a long time to be served. As is common, we choose a bundle size that is 10 times the reconfiguration latency, as this leads to a system efficiency of over 90%. For 10 Gbps links and 1500-byte packets, this size amounts to 100 packets.

5.1 Allocating resources among topologies

We allocate lasers and photodetectors among dedicated and opportunistic topologies based on the amount of traffic a ToR is expected to send or receive. Based on a day-long traffic history, we compute the maximum sending rate over a 5-minute interval for each ToR. We assume this rate is dominated by large flows [11] and the ToR will need a comparable rate in the future. Exact rate is unimportant; what matters is that historically heavy senders continue to be heavy senders; this property holds for the clusters we studied earlier.

We compute the number of lasers needed for each ToR's outgoing traffic by assuming each laser can serve 10 Gbps capacity. We bound the number thus computed by a minimum and maximum. The minimum number is 2; the maximum is the number of total lasers minus 2, to ensure each ToR has at least two dedicated lasers. We then do a similar calculation for photodetectors based on traffic received. The final number of lasers and photodetectors allocated to opportunistic links at a ToR is the maximum of the two calculations. This way, the number of opportunistic lasers equals the number of photodetectors. As some ToRs send more than they receive, an unequal allocation might appear more advantageous. However, we find empirically that it is less advantageous because it leads to imbalanced relaying capacity (incoming versus outgoing) in dedicated topologies.

A side-effect of our allocation method is that idle ToRs, which neither send nor receive heavily, have more dedicated resources. As is desirable, this property makes them a more likely relay for other ToRs in the dedicated topology.

5.2 Dedicated topology

The goal of the dedicated topology is to provide short paths for *most* flows. A number of topologies have good path length properties, such as butterflies, toruses, and random graphs. Based on recent work [36], we experimented with random graphs, but found they provide poor performance for skewed traffic matrices. It is important that ToR pairs exchanging a lot of traffic have short paths, rather than having short paths on average for all ToR pairs (most of which rarely communicate). A random graph does not distinguish among ToR pairs and their traffic demand.

We thus build a dedicated topology based on the probability of two ToRs communicating; we extract this from historical traffic matrices by dividing the traffic exchanged by total traffic. Our algorithm adds edges iteratively based on a *weighted path length* (WPL) metric. At each iteration, all possible remaining edges—between pairs of dedicated lasers and photodetectors able to connect—are considered. Each such edge changes the shortest paths between a subset of ToR pairs. We define the edge's goodness by the WPL of the resulting graph, i.e., the weighted sum of the shortest paths length between all ToR pairs, where the probability of two ToRs communicating is used as weight.

After building the dedicated topology, we compute k-shortest paths between each ToR pairs and install forwarding rules such that packets are sprayed among these paths. Our experiments use $k = 16$.

5.3 Opportunistic scheduling

Unlike dedicated links, opportunistic links are reconfigured rapidly based on current traffic. To scale to large data centers, we propose a fully decentralized and asynchronous approach. Unlike earlier approaches [16, 22, 26, 33, 38], our design does not need a centralized controller, and ToRs are not required to act at coordinated times.

The scheduling problem we have is: given a set of potential opportunistic links and current traffic bundles, find a set of active opportunistic links such that each laser is connected to at most one photodetector and vice versa. In other words, the set of active edges should form a *matching* between lasers and photodetectors.

At first blush, our problem appears as the standard switch scheduling problem: given the current state of the queues, match input-output ports. An important distinction, however, is that our problem is two-tiered. While the traffic matrix is between ToRs, matching occurs between lasers and photodetectors, and there are multiples of those per ToR.

Because of this distinction, we cannot use existing matching approaches [29, 30] out of the box. Simply stated, the presence of two-tiers complicates the computational structure of the problem. For instance, an instantaneous throughput maximizing matching for the single-tier case can be found using efficient maximum weight matching algorithms [15], but for the the two-tier case, we currently do not know whether the problem is polynomial time solvable or NP-Hard. (As we will go on to show, latency minimization is solvable in polynomial time). We could consider formulating the optimal solution as an (integer) linear program [14], but that formulation is computationally expensive and its natural implementation is centralized. Or we could force our problem to be a single-tier matching by deterministically pre-allocating bundles (e.g., using round robin) to lasers and photodetectors, but this would lower the efficiency and sacrifice the advantages of a reconfigurable system. In any case, instantaneous throughput maximization does not guarantee throughput maximization over time. Instead, we consider the objective of latency minimization.

We formulate the problem as follows. Given source ToRs S and destination ToRs D, each source $s \in S$ has a set of lasers L_s and each destination $d \in D$ has a set of photodetectors PD_d. A bundle j with source s and destination d arrives at time r_j and can be transmitted over edge $e := (l, pd)$ where $l \in L_s$ and $pd \in PD_d$, if edge e is active, i.e., if laser l is directed towards photodetector pd. At each time slot t, we must select the set of active edges and set of bundles that are transmitted over each active edge.

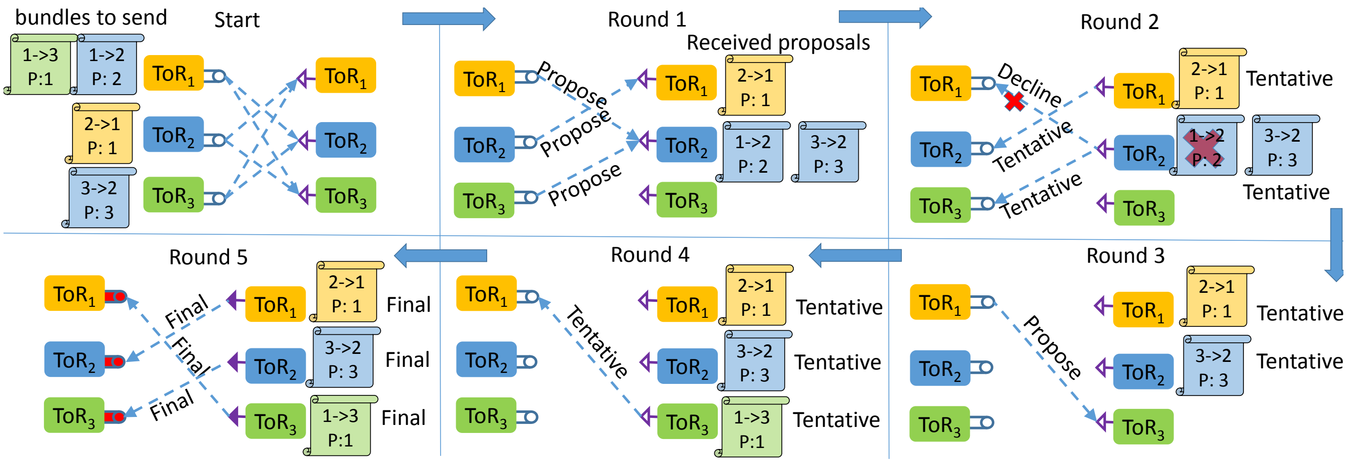


Figure 7: An example of stable matching. At start, each ToR has an interest map corresponding to outstanding bundles and their priorities (labeled by P). In top left hand corner (labeled as start), ToR_1 has a bundle to ToR_2 with priority 2 (blue bundle) and another bundle to ToR_3 with priority 1 (green bundle). In round 1, all ToRs send a proposal for their top priority bundle to the corresponding destination. The proposals are sent through the dedicated topology. In round 2, since ToR_2 received two proposals, it tentatively accepts the proposal from ToR_3 (higher priority) and declines the one from ToR_1. In the next round, ToR_1 makes a proposal for its second favorite bundle to ToR_3. In round 4, ToR_3 tentatively accepts the proposal. The algorithm ends in round 5, when there are no more proposals and all tentative arrangements become final.

The objective is to minimize latency. Specifically:

$$minimize \sum_j (c_j - r_j)^2 \qquad (1)$$

where c_j is the time when entire bundle j has been transmitted. Thus, $c_j - r_j$ is the latency of bundle j, and our objective minimizes the ℓ_2 norm of latencies of all bundles.

We solve the two-tier scheduling problem with the latency minimization objective "natively", by extending the Gale-Shapely algorithm for stable matching [17]. Our approach is constant-competitive, with a constant speed-up, against an "offline" optimal allocation that knows the entire traffic sequence. Further, unlike the conceptual frameworks used in recent work [22, 26], such as Birkhoff-von Neumann matrix decomposition [13] or Blossom-based matching [15], our algorithm is amenable to a decentralized implementation. The two-tier matching problem also arises in wired settings where multiple links exist between source-destination nodes and a switching element [27]. Our approach should be of interest in those settings as well.

In the stable matching problem, two groups, women and men, must be matched. Each person in each group has an ordered list of preferences. A matching is stable if no two people of opposite sex would both rather be matched to each other than their current partners. In our case, women and men are lasers and photodetectors.

To solve this problem via stable matching, we must assign the preferences of lasers and photodetectors for each other. We deem the preference of a laser l for a photodetector pd as the priority (defined below) of the bundle that l can transmit to pd; photodetectors' preferences are similarly based on bundles they can receive from lasers. In this setup, a stable matching implies there is no unmatched laser l, photodetector pd and bundle j such that j can be routed through l and pd, and the priority of j is higher than the priorities of both bundles that the current matching has assigned to l and pd. We set bundle priorities based on their age (i.e., $t - r_j$, where t is the current time), so that matchings able to transmit older bundles are preferred.

Our algorithm can be thought of as operating in rounds. While packets are being transmitted in the current round, it finds stable matches that will be executed in the next round. Whenever a ToR has a new bundle to send, it takes an unmatched laser that *will become* free the soonest and sends a proposal for the next upcoming round over the dedicated topology. Control packets over the dedicated topology are prioritized so they do not suffer queuing delays. Upon receiving a proposal, the destination ToR finds the lowest priority photodetector that has been tentatively matched to a proposal or is unmatched. If the newly received proposal has a higher priority, the destination rejects the previously tentative proposal, tentatively accepts the new proposal, and sends a decline message to the previous proposer. The algorithm ends when it finds matching for all proposals for the next round. Figure 7 illustrates these steps. For ease of presentation, we show a single-tier setting.

Despite its simplicity, our algorithm has the property:

THEOREM 1. *For all $\varepsilon > 0$, our stable marriage algorithm is a $2/\varepsilon + 1$-factor approximation to an optimal offline solution which knows the entire input in advance, given a speed-up of $2 + \varepsilon$, for minimizing the ℓ_2 norm of bundle latencies. This guarantee holds for all $\varepsilon > 0$ simultaneously.*

```
SENDPROPOSALS
▷ Sources periodically send proposals
while (there is a bundle to send)
  bundle ← highest priority bundle
  laser ← earliest available laser that is not matched yet
  proposal ← new Proposal()
  proposal.laser ← laser
  proposal.priority ← bundle.priority
  send proposal to bundle.dst

RECEIVEPROPOSAL(PROPOSAL)
▷ Destination received a proposal
PD ← lowest priority photodetector with status Tentative
if proposal.priority > PD.current_match.priority
  ▷ The photodetector now has a higher priority proposal
  ▷ Decline the previous tentative match
  ▷ Tentatively accept the new proposal
  proposal.photodetector ← PD
  TENTATIVEMATCH(PROPOSAL, PD) to proposal.src
  PD.status ← Tentative
  proposal.status ← Tentative
  send a decline match message to PD's previous match
  PD.current_match ← proposal
else
  send a decline match message to proposer

EXAMINEPROPOSALS
▷ Destinations periodically examine their list of proposals
for (proposal ∈ received proposals)
  if (proposal.status == Tentative &&
   all higher priority proposals have status Final)
    proposal.status ← Final
    proposal.photodetector.status ← Final
    send FINALMATCH(PROPOSAL) to proposal.src

RECEIVEFINALMATCH(PROPOSAL)
▷ Source received a Final match for a proposal
proposal.laser.status ← Final
proposal.status ← Final
if received a CTS message before timer expires
  switch the DMD image to proposal.dst
  once done, start sending the bundle
if timer expired
  mark the bundle as unmatched and release the laser

RECEIVETENTATIVEMATCH(PROPOSAL)
▷ Source received a Tentative match for a proposal
proposal.status ← Tentative
proposal.laser.status ← Tentative

RECEIVEDECLINEMATCH(PROPOSAL)
▷ Source received a Decline match for a proposal
proposal.status ← Decline
move to the next priority bundle and call SENDPROPOSALS
```

Figure 8: The event loop for two-tier and asynchronous stable matching.

The proof [14] is based on a competitive analysis that compares the cost of our algorithm to that of a hindsight optimal solution. This solution is aware of all future bundle arrivals and schedules them optimally.

Figure 8 shows the event loop, executed asynchronously at each ToR, that implements our scheduling algorithm. SENDPROPOSALS is executed at sender ToRs which, in turn, sends proposal messages to destination ToRs. Upon receipt of a proposal, RECEIVEPROPOSAL tries to tentatively match the proposal. At some point, the destination has to stop waiting for new proposals and finalize the current tentative ones. Since matching is determined while the previous round of transfers is going on, the destination has, at most, 120 μs (i.e., the time to transmit a 100-packet bundle at 10 Gbps) to finalize the matches.[1] We set an event at destinations to call EXAMINEPROPOSALS every 80 μs to finalize tentative proposals and send a final match message to proposers. Upon receipt of this message, the source will mark the bundle as final-matched. Since matching is happening while the previous round of data is being transmitted, the photodetector will inform its next matched laser by sending a clear-to-send (CTS) message once the current transmission ends. The transmission starts right away or as soon as the laser finishes its own current transmission (and after loading the CGP to the DMD, if needed). Bundles arriving in the middle of a transmission are considered for scheduling when SENDPROPOSALS is executed next.

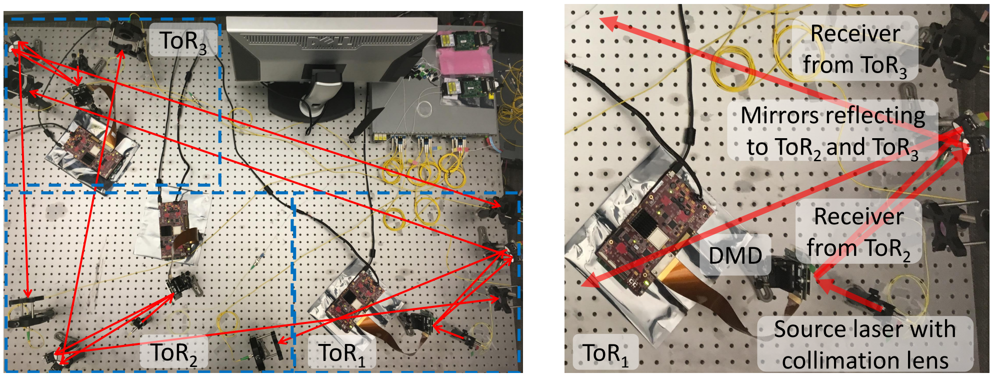
To ensure against failures, each finalized match has a timeout of 100 μs to prevent deadlocks. If the timeout expires, the matching becomes invalid, and the bundle and its matched laser and photodetector go back to the pool of unmatched resources.

6. PROTOTYPE

We evaluate ProjecToR using a small-scale prototype and large-scale simulations. The two experimental frameworks permit us to study different aspects of ProjecToR. The prototype allows us to benchmark switching time, throughput of free-space transmissions using DMDs, and power loss. The simulations allow us to study application performance with realistic traffic patterns. This section focuses on the prototype; and the next one on simulations.

We built a three-ToR prototype of ProjecToR using three Texas Instruments DLP Discovery 4100 kits with 0.7 XGA Chipset [2]. Each ToR is equipped with one transmitter and one DMD. The layout of this prototype is shown in Figures 9(a) and 9(b). Instead of using independent lasers and photodetectors, we use components embedded in commodity transceivers. We emulate a ProjecToR transmitter using modulated light coming out of a commodity transceiver [3] with 1550 nm wavelength and 4 dBm launch power, that directs the light to the DMD, marked as source laser in Figure 9(b). The DMD diffracts the light into free-space towards the receiver. We emulate a receiver by injecting the incoming free-space light into fiber, after which the light goes back to a commodity transceiver at the receive side. A collimation lens [1] with insertion loss less than 0.2 dB is used at the sender side, to keep the output beam diameter constant and prevent the light from diverging as it travels towards the

[1] At higher speeds, the bundle size should increase. We leave the evaluation of larger bundle sizes to future work.


ToR_3
ToR_2
ToR_1
Receiver from ToR_3
Mirrors reflecting to ToR_2 and ToR_3
Receiver from ToR_2
DMD
Source laser with collimation lens
ToR_1

(a) ProjecToR prototype

(b) Zoomed into bottom right corner

Figure 9: A three-ToR ProjecToR interconnect. The red lines illustrate the path traveled by light in free-space.

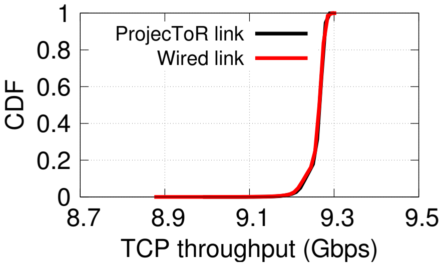


Figure 10: TCP throughput comparison between ProjecToR link and wired link (the lines are overlapping).

	Measured loss (dB)
Laser to collimating lens	0.01
DMD	10.41
Aperture	0.00
Mirror assembly	0.01
Focusing lens	0.01
Receiver	0.05 (6)
End-to-end	10.49 (16.44)

Table 2: Optical loss measured at each component.

destination. For stability, the prototype was mounted on an optical table [4]. We do all optical alignments manually.

Our approach of using commodity transceivers is driven by pragmatic concerns. It allows us to exchange data without needing separate modulation logic at the transmitter or demodulation logic at the receiver. Application packets are forwarded to the transceiver which modulates the light for us. However, this method has limitations as well. First, injecting light back into the fiber at the receiver, rather than using an exposed photodetector as would be the case in the actual system, introduces power loss (see below). We use amplifiers in our prototype to overcome this loss. Second, commodity transceivers can take up to a second, after receiving light, to report that the link is up. They are designed for a fiber-based world where links are not expected to toggle rapidly. The switching speed for *data* transmissions is dominated by this delay in our prototype. However, we benchmark how quickly the DMD changes the direction of the modulated *light* using an oscilloscope (see below).

We divide our prototype experiments into two parts. In §6.1, we micro-benchmark a ProjecToR free-space link using various relevant metrics. Then, in §6.2, we study the ProjecToR scheduler and demonstrate its advantage over a Firefly-style interconnect.

6.1 Micro-benchmarks

We benchmark several key properties of ProjecToR's DMD-based FSO links: *i*) does diffraction through the DMD over free-space impact link throughput? *ii*) what is the end-to-end power loss? *iii*) what is the switching speed of the DMD? and *iv*) can the diffracted light travel long distances without loss in intensity?

Throughput: To study if our free-space links deliver the same throughput as a wired link, we run TCP iPerf between ToR_1 and ToR_2 in Figure 9(a), first connected via free-space and then via fiber. Each run lasted a day, and we measured the resulting TCP throughput over 10-second intervals. Figure 10 compares the CDF of throughput of the two runs. The throughput is the same in both cases, and we do not observe any packet drops or packet errors for either.

Power budget: Optical power loss is a key measure for any optical system. Table 2 shows the power loss measured at each component of a ProjecToR link. Attenuation of light traveling in free-space—0.2 dB/km for 1550 nm wavelength [24]—is negligible given our distances of 100s of meters. We report two numbers for receiver, depending on whether the received light is injected to photodetector (0.05 dB) or fiber (6 dB). Recall that in our prototype, fiber injection is needed for us to use commodity transceivers, but the actual system will use photodetectors. The total loss is 10.49 dB when photodetector injection is used and it is 16.44 dB when injecting the light to fiber.

DMD introduces the most loss, of 10.41 dB. Multi-level DMDs have a loss of 0.7 dB [39] but they are not yet commercially available. When they are, the optical loss of a ProjecToR link will be under 1 dB.

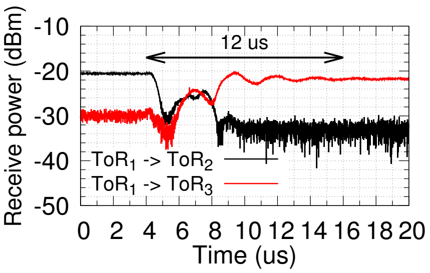


Figure 11: Loss-of-light time measured when switching.

Switching time: To measure the switching time of the DMD, in the setup of Figure 9(a), we replace the transmitter at ToR_1 with a -10 dBm laser and the receivers at ToR_2 and ToR_3 with InGaAs PIN photodetectors. We then switch light between the two photodetectors by loading different CGPs on the DMD at ToR_1. With the two photodetectors connected to an oscilloscope, we can measure instantaneous light intensity and accurately measure switching speed.

Figure 11 shows what happens as we switch light. The switching time is the loss-of-light time—the time between when the first receiver loses the signal and the second one gets stable light intensity above its sensitivity threshold (which in our case is -24 dBm). Initially, ToR_2 is receiving light. Then, at time $t = 4$ μs, the switching command is fired. After 10 μs, i.e., $t = 14$ μs, ToR_3 has stable light that is above the receiver's sensitivity level. This experiment micro-benchmarks the DMD's switching time, and additional time might be required for transceivers to start decoding bits. This additional time is not fundamental, however, and prior work has been able to mitigate it [33]. Conservatively, we consider 12 μs as the switching time in our later experiments.

Long distance: Next, we evaluate if traveling long distances reduces light intensity at the receiver. Such a reduction can occur if the light beam coming off the DMD of the post reflection from the mirror is not cylindrical but expands, such that only a fraction falls on the photodetector.

For this experiment, we place one prototype in a long hallway and keep another photodetector nearby. We make light travel over 100 meters by directing the laser light to hit a mirror placed 50 meters away at the end of the hallway. We then switch the light between the two photodetectors and monitor the light intensity at far and near photodetectors. Both photodetectors turn out to have similar power levels as the DMD switches between the two. We omit detailed results due to space constraints.

6.2 ProjecToR scheduling in action

We now illustrate the behavior of the ProjecToR scheduler in our prototype. In this experiment, a separate, wired network acts as the dedicated topology, and the controller uses this network for stable-matching control messages as well. Opportunistic traffic demand is such that each ToR always has bundles to send to each of the other two ToRs. A new bundle for the destination is generated as soon as one is delivered. Bundles are transferred over the free-space links, per the scheduling algorithm in §5.3. Figure 12(a) shows the resulting opportunistic transfers. Each notch denotes a bundle transferred from a source to a destination ToR.

For contrast, Figure 12(b) shows the behavior that emerges when laser-photodetector matching is forced to be symmetric (e.g., as in FireFly). In this case, lasers and photodetectors are coupled (as in a transceiver); when the laser at ToR_1 transmits to the photodetector at ToR_2, the laser at ToR_2 can only transmit to the photodetector at ToR_1. We find that ProjecToR's ability to establish asymmetric links (as shown in Figure 12(a)) results in 45% higher throughput because it allows for more flexible configurations (e.g., those that can support asymmetric traffic demands).

7. SIMULATIONS

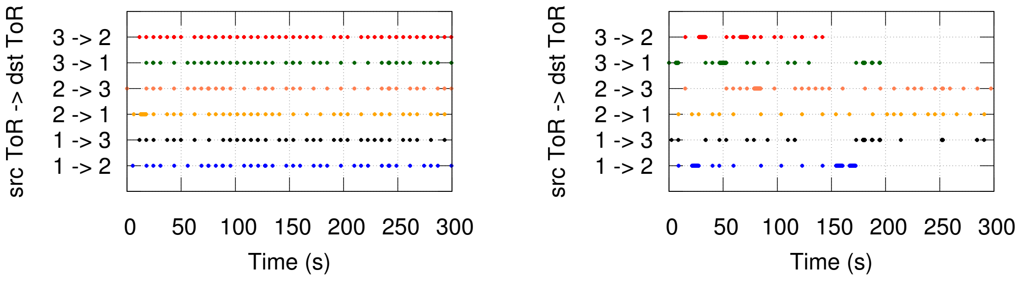
We now seek to understand the behavior of a large scale ProjecToR interconnect compared to that of static and other reconfigurable interconnects. We choose full-bisection bandwidth fat tree with k-port switches [10] as a benchmark for static topology. We use FireFly as the state of art method for reconfigurable interconnects. Like ProjecToR, it provides seamless reconfigurability based on FSO (but it lacks high fan-out and reconfiguration agility). In this section:

1. We provide a cost analysis of ProjecToR; which informs our simulations such that we are comparing interconnects of roughly similar (but not identical) costs.
2. We demonstrate the overall performance of ProjecToR using packet level simulations and show that it suits today's data center traffic better than fat tree or FireFly.
3. We show ProjecToR's seamless reconfigurability, high fan-out, and low switching time are key to performance

Background on FireFly: FireFly is a reconfigurable interconnect that provisions FSO-based transceivers on top of racks. Each transmitter has a fan-out of 10, enabled by switchable/Galvo mirrors with a switching time of 20 ms. Unlike ProjecToR, FireFly has no dedicated-opportunistic split of resources, and FSO links are bi-directional. The routing design of FireFly divides time into reconfiguration epochs. Based on the current traffic matrix, a centralized controller decides which FSO links should be formed for the next epoch, using an extended version of the Blossom [15] algorithm. Packets are routed over multiple hops and spread along multiple paths, with the relative fractions computed by solving a multi-commodity flow problem.

7.1 Cost analysis

We estimate the costs of a full-bisection fat tree, FireFly, and ProjecToR. A fat tree with k-port switches has k^3 switch-to-switch cables and transceivers, $k^2/4$ core switches, and k^2 pod switches. We provision a comparable ProjecToR interconnect with k lasers, k photodetectors, k DMDs, and k mirror assemblies per ToR without any cables or intermediate switches. We provision FireFly similarly, with k FSO links, and up to 10 switchable/Galvo mirrors. The ToRs at all three interconnects have $k/2$ ports to connect to servers. For fat tree cables, we use average length of 300 meters.



(a) Asymmetric connectivity (ProjecToR)

(b) Symmetric connectivity (FireFly-like)

Figure 12: Opportunistic transfers in our prototype.

Component	Cost ($)

ProjecToR Tx+Rx components	80(low)
	180(high)

DMD	100
Mirror Assembly+Lens	50
SR transceiver	80
Optical cable/meter	0.3
ToR port	90
Galvo mirror	200

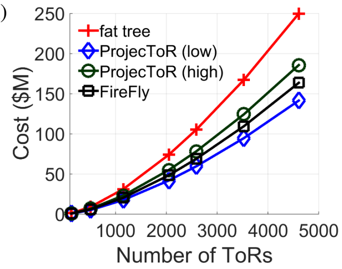


Figure 13: Component and interconnect costs.

The graph in Figure 13 (right) shows the cost of the three interconnects for a 10 Gbps deployment, using the component costs in Figure 13 (left). Consistent with earlier analysis [22], this figure suggests a FireFly interconnect is about 35% cheaper than fat tree.

For ProjecToR we provide high and low estimates as follows. The electrical-optical transmit and receive components in ProjecToR (e.g., lasers, photodetectors, modulators) are the same as those in today's transceivers. We thus estimate their total price based on transceiver cost. Our high-end estimates are based on long reach (LR) transceivers, to compensate for the power budget numbers reported in §6. LR transceivers have up to 15 dB power budget [6]. Our low-end estimates are based on short reach (SR) transceivers [7]. A commodity 2-level DMD retails for $100 [9], and we conservatively estimate the mirror assembly and lenses will cost $50 at scale. These estimates bring the total cost of a ProjecToR interconnect to 25–40% lower than full-bisection bandwidth fat tree. Our estimate might not account for unanticipated costs given the radical departure that ProjecToR represents. Hence, we are allowing a large cost margin. Even so, as the following sections show, ProjecToR outperforms fat tree and FireFly.

7.2 Simulation methodology

We evaluate ProjecToR using a custom packet-level simulator. Our simulations use traffic traces from clusters introduced in §2. These traces contain information on bytes transferred in 5-minute intervals but lack information on flow arrivals and sizes. We first convert bytes-transferred data into the probability of communication between two ToRs. We then generate TCP flows with a Poisson arrival rate λ/s (see below). The size of a flow is based on distributions studied in prior work [11], and its source and destination endpoints are based on the computed ToR-pair communication probabilities. Finally, we use the traffic from the previous day to inform the allocation of lasers and photodetectors among dedicated and opportunistic topologies and the construction of dedicated topology. To aid reproducibility of our results, our traffic trace data are publicly available [5].

In our experiments, we use λ as a knob to tune the level of load on the network. We increase λ until the average utilization of the most congested link in the fat tree topology is 80%. We find that beyond this load, fat tree is unable to finish a substantial fraction (over 5%) of flows. In each experiment, average load represents the back calculated average utilization of the most congested link.

We use a 128 node topology ($k = 16$) as the basis for our simulations. With $k = 16$, this topology provides a good benchmark to examine dedicated topology construction, stable matching algorithm and fan-out requirements. We select $Cluster2$ and $Cluster4$ as representative clusters because they have different size and communication patterns and randomly select 128 ToRs from them. We assume a reconfiguration latency of 20 *ms* for FireFly and 12 μs for ProjecToR. We use a bundle size of 100 packets for ProjecToR, hence, reconfiguring its opportunistic links every 120 μs.

Below, we first demonstrate that ProjecToR achieves up to 95% better end-to-end performance with workloads observed in deployed data centers (§7.3). We then deconstruct the overall results and show how agility and fan-out contribute to its performance (§7.4).

7.3 Overall performance comparison

We consider four interconnects from different classes:

1. Full-bisection bandwidth fat tree [10]: not reconfigurable.
2. FireFly: seamless reconfiguration, but with low agility and low fan-out.
3. ProjecToR: seamless reconfiguration, with high agility and high fan-out.
4. Partitioned ProjecToR: a reconfigurable topology where a pre-determined portion of links are connected to the opti-

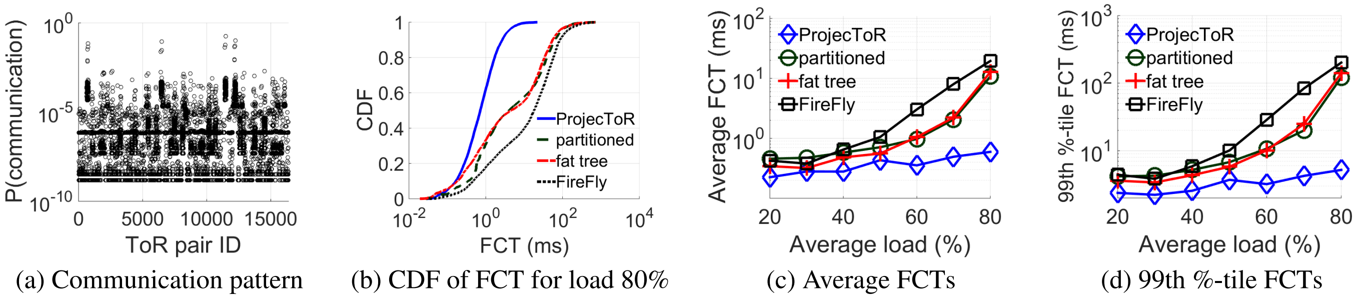


Figure 14: Flow completion times (FCT) for traffic matrix from *Cluster2* using packet level simulator.

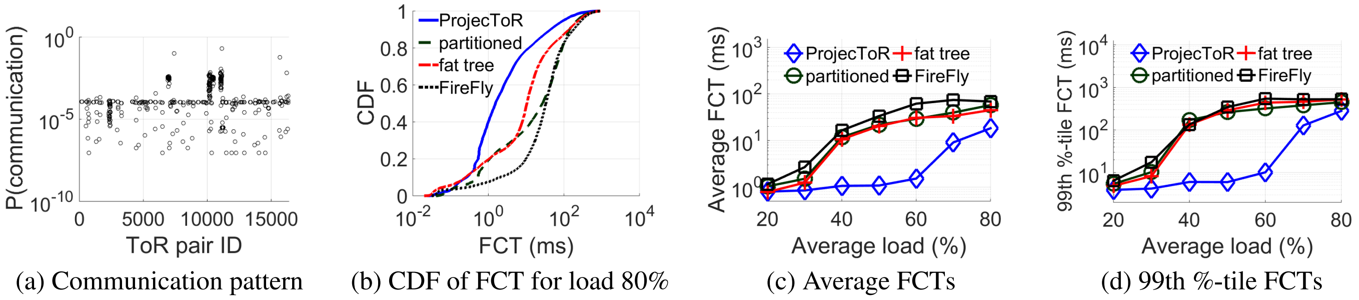


Figure 15: Flow completion times (FCT) for traffic matrix from *Cluster4* using packet level simulator.

cal switch and the remaining links are connected via electrical packet switches. Examples of such interconnects include Helios [16], c-through [38], and Solstice [26]. We emulate their approach by statically partitioning half the lasers for dedicated links and half for opportunistic links. This way, the interconnect inherits the high agility and fan-out of ProjecToR, and any performance differences can be attributed to its partitioned approach.

Figure 14(a) shows a random snapshot of communication patterns between ToR pairs selected from *Cluster2*. Most ToR pairs are active, although some pairs are more likely to communicate with each other than others. In contrast, Figure 15(a) shows communication patterns in *Cluster4*; only a few ToR pairs are communicating, leaving the rest of the cluster more or less quiet.

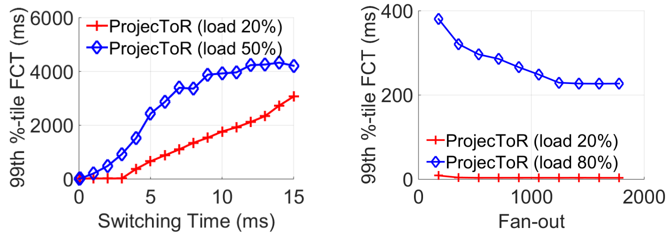
Let us begin with *Cluster2*. We use flow completion time (FCT) as the key metric to evaluate each interconnect. Figure 14(b) shows the CDF of FCT when load is 80%. The median FCT is an order of magnitude lower in ProjecToR than in fat tree. Partitioned ProjecToR (labeled as partitioned) and fat tree have similar FCTs. FireFly has the worst performance. [2] Even though FireFly can seamlessly reconfigure network capacity, it has poorer performance than ProjecToR because of its low agility—20 *ms* reconfiguration time—and low fan-out.

Figures 14(c) and (d) plot the average and 99^{th}%-tile FCT across different loads. We observe a consistent trend in the relative performance of interconnects. Recall that the load parameter indicates average load of the highest congested link in fat tree topology with the given communication probabilities. A full-bisection bandwidth fat tree should be able to support close to 100% load if the communication probabilities are uniform and random. However, because traffic pattern is skewed, buffers at the congested links fill up quickly, resulting in high FCTs. Because ProjecToR adjusts the topology quickly, it achieves low FCTs. Across all loads, it reduces average FCT by 30–95% compared to fat tree.

We now repeat the same analysis for *Cluster4*. Figure 15(b) shows an interesting trend in FCTs when average load is 80%. Most ProjecToR flows have smaller FCTs compared to all other interconnects except at the very tail. This is because at such a high load, and with such a skewed traffic matrix, the bulk of traffic is between a few heavy ToR pairs, and all opportunistic links in ProjecToR are close to saturation; hence, transient congestion creates high tail FCT.

As it turns out, this traffic pattern starts impacting FCTs at even lower average loads; see Figure 15(c) and (d). Up to 30–70% average load, ProjecToR is able to mitigate the impact of heavy transfers by provisioning opportunistic links and, hence, reducing average FCT by 30–95% . When load is increased to 80%, opportunistic capacity becomes close to saturation; ProjecToR's average FCT is still 60% better than

[2] Hamedazimi et al. [22] also compare FireFly and fat tree networks. Our results agree in some aspects (e.g., the performance of FireFly drops when the traffic matrix is less skewed) and disagree in others (e.g., FireFly tends to have worse performance than fat tree). We could not obtain FireFly's simulation source code to cross check the differences.



(a) Varying switching time (b) Varying fan-out

Figure 16: Impact of reconfiguration time and fan-out.

that of fat tree, but the tail FCT is closer to fat tree because of bursts and transient congestion.

Finally, we can see in Figures 14 and 15 that, for both clusters, ProjecToR outperforms its partitioned counterpart. This advantage stems directly from its seamlessness as all other aspects of the two interconnects are identical.

7.4 Impact of fan-out and switching time

We now dig deeper into ProjecToR's performance and show that, like seamlessness, high agility and high fan-out are important factors for the performance of a reconfigurable interconnect. For these experiments, we change ProjecToR switching time and fan-out and study the impact on performance. We use 99^{th}%-tile FCT as the performance metric; results are similar for average FCT.

In Figure 16(a), we increase the switching time from 10 μs to 15 ms. We study two loads levels: 50% and 20%. In both cases, FCTs rapidly increase beyond 1 second as switching time increases. The reason is tied to the cost of switching on FCTs. Simply stated, packets need to wait longer to be scheduled. Our results show that more than a few milliseconds switching time significantly degrades performance for real workloads. When the switching time is in microseconds, FCTs are small and thus overlap visually near zero in the graph. But the data show that higher switching times have higher FCTs.

Figure 16(b) demonstrates the impact of fan-out on performance. In our experiments we have 128 racks, each with at most 14 opportunistic lasers and 14 opportunistic photodetectors. The maximum fan-out required by a transmitter at a rack is the number of photodetectors a laser can reach, i.e., 1778 (127×14). This limit is the right-most point on the x-axis. The left-most point is 127, which represents a transmitter's ability to reach only one photodetector per rack. Intermediate points represent different numbers of photodetectors per rack that a transmitter can reach.

The figure shows performance decreases as fan-out decreases. In each fan-out case, the same number (14) of maximum parallel opportunistic links can be established between two ToRs—14 different lasers can point at 14 different photodetectors at a remote rack even when the fan-out is minimum (127). What changes is the flexibility that results from the ability to reach additional photodetectors at remote racks. If one reachable photodetector is busy (because of another transmitter), it helps to be able to reach another one. Our matching algorithm exploits this flexibility. The importance of high fan-out is more pronounced at high loads where there is more competition at both lasers and photodetectors.

8. RELATED WORK

Our work is inspired by prior works on reconfigurable DC interconnects. Researchers have explored the use of several underlying technologies to build such networks. Helios, Mordia, and Reactor use optical circuit switches [16,25,33]. OCS-based approaches suffer from limited fan-out and slow switching time; in contrast, ProjecToR can easily support tens of thousands of ports. Flyways and Zhou et al. use 60GHz wireless technology [23,40]. Unlike optical technology, RF tends to suffer from limited throughput and interference and thus cannot scale to large DCs.

Firefly [22] is similar to ProjecToR in its use of FSO, but it uses Galvo or switchable mirrors as the basis for switching. In contrast, we use DMDs, which enable a single transmitter to reach thousands of receivers, as opposed to 10 for Firefly. They also enable 7-12 μs switching time between receivers, as opposed to 20 ms for Firefly.

Our second source of inspiration comes from the work of Miles et al. [32] and Lynn et al. [28], which explore the use of DMDs for optical switching. Their exploration is in the context of building an all optical switch whereas our exploration is in the context of an FSO-based interconnect, which introduces a whole new set of challenges. Some of these challenges are the covering of a large space, increasing the reach of transmitters to thousands of receivers, and the scheduling of transmissions over a large interconnect.

9. CONCLUSION

We explored a novel way to build a DC interconnect that allows each rack to establish direct links to each other rack and reconfigure such links within 12 μs. We showed how to effectively use such a flexible interconnect and developed an online flow scheduling algorithm that has provable guarantees. Our experiments and analysis indicate that our approach can improve mean flow completion times by 30-95%, while reducing cost by 25–40%.

We are the first to admit that we are proposing a radical departure from the current norms and have barely scratched the surface in terms of fully developing the approach and demonstrating its practicality. Of particular concern is how physical properties of data centers (e.g., vibration, dust, and humidity) will hinder the performance of FSO links. We are investigating these issues in ongoing work.

Acknowledgments We thank our colleagues Victor Bahl, Andreas Georgiou, Srikanth Kandula, Joel Kollin, Brittany Lynn, Doug Patterson, Ajay Gautham Manicka Sureshbabu, and Devin Thorne for sharing their expertise and helping execute this work. We also thank our shepherd Alex Snoeren and the SIGCOMM reviewers for comments on the paper.

10. REFERENCES

[1] Collimation lens. http://www.thorlabs.us/thorproduct.cfm?partnumber=CFS18-1550-APC.

[2] DLP discovery 4100 development kit. http://www.ti.com/tool/dlpd4x00kit.

[3] Multirate 80km SFP+ optical transceiver. https://www.finisar.com/optical-transceivers/ftlx1871m3bcl.

[4] Optical table with active isolator legs. http://www.thorlabs.us/newgrouppage9.cfm?objectgroup_id=5930.

[5] ProjecToR home page. http://research.microsoft.com/en-us/projects/projector.

[6] SFP+ 10km reach transceiver. http://www.robofiber.com/content/datasheets/SFP-1010-LR-datasheet.pdf.

[7] Sr 300m multi-mode 10g transceiver. http://www.robofiber.com/sfp-1000-sr.

[8] Texas Instruments DLP technology overview. http://www.ti.com/lsds/ti/analog/dlp/overview.page.

[9] Texas instruments store. https://store.ti.com/Search.aspx?k=dlp&pt=-1.

[10] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. *SIGCOMM'08*, pages 63–74.

[11] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker. pFabric: Minimal near-optimal datacenter transport. *SIGCOMM'13*, pages 435–446.

[12] L. Benjamin. DMD 101: Introduction to digital micromirror device (DMD) technology. *Texas Instruments*, Oct. 2013.

[13] D. Birkhoff. Tres observaciones sobre el algebra lineal. *Universidad Nacional de Tucuman Revista , Serie A*, 5:147–151, 1946.

[14] N. Devanur, J. Kulkarni, G. Ranade, M. Ghobadi, R. Mahajan, and A. Phanishayee. Stable matching algorithm for an agile reconfigurable data center interconnect. Technical Report MSR-TR-2016-34, 2016.

[15] J. Edmonds and E. L. Johnson. Matching, Euler tours and the chinese postman. *Mathematical Programming*, pages 88–124, 1973.

[16] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat. Helios: A hybrid electrical/optical switch architecture for modular data centers. *SIGCOMM'10*, pages 339–350.

[17] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[18] A. Georgiou, J. Christmas, J. Moore, A. Jeziorska-Chapman, A. Davey, N. Collings, and W. A. Crossland. Liquid crystal over silicon device characteristics for holographic projection of high-definition television images. *Appl. Opt. 2008*.

[19] R. W. Gerchberg and W. O. Saxton. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik 35*, 237, 1972.

[20] M. Ghobadi, R. Mahajan, A. Phanishayee, P.-A. Blanche, H. Rastegarfar, M. Glick, and D. Kilper. Design of mirror assembly for an agile reconfigurable data center interconnect. Technical Report MSR-TR-2016-33, 2016.

[21] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: A scalable and flexible data center network. *SIGCOMM'09*, pages 51–62.

[22] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer. Firefly: A reconfigurable wireless data center fabric using free-space optics. *SIGCOMM'14*, pages 319–330.

[23] S. Kandula, J. Padhye, and P. Bahl. Flyways to de-congest data center networks. *HotNets'09*, 2009.

[24] I. I. Kim, B. McArthur, and E. J. Korevaar. Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications. *Proc. SPIE*, 4214:26–37, 2001.

[25] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter. Circuit switching under the radar with REACToR. *NSDI'14*, pages 1–15.

[26] H. Liu, M. K. Mukerjee, C. Li, N. Feltman, G. Papen, S. Savage, S. Seshan, G. M. Voelker, D. G. Andersen, M. Kaminsky, G. Porter, and A. C. Snoeren. Scheduling techniques for hybrid circuit/packet networks. *CoNext'15*.

[27] Y. J. Liu, P. X. Gao, B. Wong, and S. Keshav. Quartz: A new design element for low-latency dcns. *SIGCOMM'14*, pages 283–294.

[28] B. Lynn, P.-A. Blanche, A. Miles, J. Wissinger, D. Carothers, L. LaComb, R. Norwood, and N. Peyghambarian. Design and preliminary implementation of an $N \times N$ diffractive all-optical fiber optic switch. *Journal of Lightwave Technology*, 31(24):4016–4021, Dec 2013.

[29] N. McKeown. The iSLIP scheduling algorithm for input-queued switches. *IEEE/ACM Trans. Netw.*, 7(2):188–201, Apr. 1999.

[30] N. Mckeown, B. Prabhakar, and M. Zhu. Matching output queueing with a combined input output queued switch. *IEEE Journal on Selected Areas in Communications*, pages 1030–1039, 1999.

[31] W. Mellette and J. E. Ford. Scaling limits of free-space tilt mirror mems switches for data center networks. *Optical Fiber Communication Conference*, page M2B.1, 2015.

[32] A. Miles, B. Lynn, P. Blanche, J. Wissinger, D. Carothers, A. LaComb Jr., R. Norwood, and N. Peyghambarian. 7×7 DMD-based diffractive fiber switch at 1550 nm. *Optics Communications*, 334:41–45, Jan 2015.

[33] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat. Integrating microsecond circuit switching into the data center. *SIGCOMM'13*, pages 447–458.

[34] Y. K. Rabinovitz. Digital Light Processing Technology (DLP) Beyond any conventional projection. http://www.opli.net/magazine/eo/2011/news/dlp_tech.aspx, 2011.

[35] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social network's (datacenter) network. *SIGCOMM'15*, pages 123–137.

[36] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. *NSDI'12*, pages 225–238.

[37] A. Singla, A. Singh, and Y. Chen. OSA: An optical switching architecture for data center networks with unprecedented flexibility. *NSDI'12*, pages 239–252.

[38] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan. c-Through: Part-time optics in data centers. *SIGCOMM'10*, pages 327–338.

[39] B.-W. Yoo, M. Megens, T. Sun, W. Yang, C. J. Chang-Hasnain, D. A. Horsley, and M. C. Wu. A 32×32 optical phased array using polysilicon sub-wavelength high-contrast-grating mirrors. *Opt. Express*, 22(16):19029–19039, Aug 2014.

[40] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H. Zheng. Mirror mirror on the ceiling: Flexible wireless links for data centers. *SIGCOMM'12*, pages 443–454.