Breaking the VLB Barrier for Oblivious Reconfigurable Networks

Tegan Wilson
Cornell University
Ithaca, New York, USA

Daniel Amir
Cornell University
Ithaca, New York, USA

Nitika Saran
Cornell University
Ithaca, New York, USA

Robert Kleinberg
Cornell University
Ithaca, New York, USA

Vishal Shrivastav
Purdue University
West Lafayette, Indiana, USA

Hakim Weatherspoon
Cornell University
Ithaca, New York, USA

ABSTRACT

In a landmark 1981 paper, Valiant and Brebner gave birth to the study of oblivious routing and, simultaneously, introduced its most powerful and ubiquitous method: *Valiant load balancing (VLB)*. By routing messages through a randomly sampled intermediate node, VLB lengthens routing paths by a factor of two but gains the crucial property of *obliviousness*: it balances load in a completely decentralized manner, with no global knowledge of the communication pattern. Forty years later, with datacenters handling workloads whose communication pattern varies too rapidly to allow centralized coordination, oblivious routing is as relevant as ever, and VLB continues to take center stage as a widely used — and in some settings, provably optimal — way to balance load in the network obliviously to the traffic demands. However, the ability of the network to rapidly reconfigure its interconnection topology gives rise to new possibilities.

In this work we revisit the question of whether VLB remains optimal in the novel setting of reconfigurable networks. Prior work showed that VLB achieves the optimal tradeoff between latency and *guaranteed* throughput. In this work we show that a strictly superior latency-throughput tradeoff is achievable when the throughput bound is relaxed to hold with high probability. The same improved tradeoff is also achievable with guaranteed throughput under time-stationary demands, provided the latency bound is relaxed to hold with high probability and that the network is allowed to be *semi-oblivious*, using an oblivious (randomized) connection schedule but demand-aware routing. We prove that the latter result is not achievable by any fully-oblivious reconfigurable network design, marking a rare case in which semi-oblivious routing has a provable asymptotic advantage over oblivious routing. Our results are enabled by a novel oblivious routing scheme that improves VLB by stretching routing paths the minimum possible amount — an additive stretch of 1 rather than a multiplicative stretch of 2 — yet still manages to balance load with high probability when either the traffic demand matrix or the network's interconnection schedule are shuffled by a uniformly random permutation. To analyze our routing scheme we prove an exponential tail bound which may be of independent interest, concerning the distribution of values of a bilinear form on an orbit of a permutation group action.

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649608

CCS CONCEPTS

• **Theory of computation → Network flows**; • **Mathematics of computing → Probabilistic algorithms**; • **Networks → Network algorithms**.

KEYWORDS

Oblivious routing, reconfigurable networks, tail inequalities, valiant load balancing

ACM Reference Format:
Tegan Wilson, Daniel Amir, Nitika Saran, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. 2024. Breaking the VLB Barrier for Oblivious Reconfigurable Networks. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3618260.3649608

1 INTRODUCTION

Reconfigurable networks use rapidly reconfiguring switches to create a dynamic time-varying topology, allowing for great flexibility in efficiently routing traffic. This idea has gained prominence due to recent technologies such as optical circuit switching [15, 39] and free-space optics [17, 23, 42] that enable reconfigurations within microseconds [28, 32] or even nanoseconds [11, 12]. Datacenter network architectures that leverage this capability are now being actively explored, including with recent prototype systems [7, 18, 29, 35] and theoretical modeling and analysis [1, 3, 40]. The rate of change of datacenter network workloads (summarized by a time-varying traffic demand matrix) has already outpaced the reconfiguration speeds achievable using a central controller [18], driving researchers to focus on **oblivious reconfigurable networks (ORNs)**, which use a *demand-oblivious* reconfiguration and routing mechanism that is fully decentralized.

An analogous set of questions came to the fore in an earlier era of computing research, when the focus was on designing communication schemes for parallel computers. The network model at that time — a fixed, bounded-degree topology — was very different, but the objective was the same: to efficiently simulate arbitrary communication patterns among a set of N nodes without requiring any centralized control. In a landmark 1981 paper, Valiant and Brebner articulated the central problem in terms that still resonate with the practice of modern datacenter networking.

> *The fundamental problem that arises in simulating on a realistic machine one step of an idealistic computation is that of simulating arbitrary connection patterns among the processors via a fixed sparse network... For routing the packets the strategy will have to be based on only a minute fraction of the total information necessary to specify the complete communication pattern.*

The solution proposed by Valiant and Brebner, which henceforth came to be known as *Valiant load balancing* or *VLB*, was beautifully simple: to send data from source s to destination t, sample an intermediate node u uniformly at random. Then form a routing path from s to t by concatenating "direct paths" from s to u and from u to t. (The definition of direct paths may depend on the network topology; often shortest paths suffice.) This lengthens routing paths by a factor of two and thus consumes twice as much bandwidth as direct-path routing. However, crucially, it is *oblivious*: the distribution over routing paths from s to t depends only on the network topology, not the communication pattern. Oblivious routing schemes satisfy the desideratum of being "based on only a minute fraction of the total information necessary to specify the complete communication pattern" in the strongest possible sense.

The focus of oblivious routing research in the 1980's was on network topologies designed to enable efficient communication among a set of processors. These topologies, such as hypercubes and shuffle exchange networks, tended to be highly symmetric (often with vertex- or edge-transitive automorphism groups) and tended to have low diameter and no sparse cuts. One could loosely refer to this class of networks as *optimized topologies*. A second phase of oblivious routing research, initiated by Räcke in the early 2000's, designed oblivious routing schemes for *general topologies*. Compared to optimized topologies, the oblivious routing schemes for general topologies require much greater overprovisioning, inflating the capacity of each edge by at least a logarithmic factor compared to the capacity that would be needed if routing could be done using an optimal (non-oblivious) multicommodity flow. The construction of oblivious routing schemes with polylogarithmic [9, 24, 33] and eventually logarithmic [34] overhead was a seminal discovery for theoretical computer science, but did not improve over the performance of VLB for optimized topologies.

Remarkably, more than 40 years after the introduction of VLB, it remains the state of the art for oblivious routing in optimized topologies. In fact, existing results in the literature show that the factor-of-two overprovisioning associated with VLB is optimal in at least two important contexts: when building a network of fixed-capacity links to permit any communication pattern with bounded ingress and egress rates per node [6, 26, 41], and when designing an oblivious reconfigurable network with bounded maximum latency, again to permit any communication pattern with bounded ingress and egress rates per node [3]. In both cases, authors proposed optimized topologies, analyzed routing protocols which use VLB, and provided lower bounds that matched the VLB performance.

Running the network is responsible for a significant fraction of the cost of modern datacenters. The capital cost of the networking equipment alone accounts for around 15% of the total cost to build and run a datacenter; this increases to over 30% when including indirect costs such as power and cooling for network equipment [8, 19]. Overprovisioning the network increases these costs proportionally [35], which motivates investigating when it is possible to "break the VLB barrier" and reap the benefits of oblivious routing without paying the cost of provisioning twice as much capacity as needed for optimal demand-aware routing.

In this work we show that *the ability to randomize the network topology in reconfigurable networks indeed allows oblivious routing schemes that break the VLB barrier*. We present a novel oblivious routing scheme for reconfigurable networks with a randomized connection schedule. The routing paths used by our scheme exceed the length of shortest (latency-bounded) paths by the smallest possible amount: *an additive stretch of 1 rather than a multiplicative stretch of 2*. Building upon this new routing scheme, we obtain reconfigurable network designs that improve the throughput achievable within a given latency bound by nearly a factor of two, under two relaxations of obliviousness:

(1) when the network is allowed a small probability of violating the throughput guarantee; or
(2) when the throughput guarantee must hold with probability 1, but routing is only *semi-oblivious*.

Semi-oblivious routing refers to routing schemes in which the network designer must pre-commit (in a demand-oblivious manner) to a limited set of routing paths between every source and destination, but the decision of how to distribute flow over those paths is made with awareness of the requested communication pattern. In the context of reconfigurable networks, this means that the connection schedule is oblivious but the routing scheme may be demand-aware. In fact, the semi-oblivious routing scheme that we refer to in Result 2 above is demand-aware in a very limited sense: it uses the oblivious routing scheme from Result 1 with high probability, but in the unlikely event that this leads to congestion on one or more edges, it reverts to using a different oblivious routing scheme that is guaranteed to avoid congestion at the cost of incurring higher latency. Note that this semi-oblivious routing scheme only requires network nodes to share one bit of common knowledge about the communication pattern (namely, whether or not there exists a congested edge), hence it still obeys Valiant and Brebner's desideratum that routing decisions are based on only a minute fraction of the total information needed to specify the communication pattern. In the full version of our paper, we prove that purely oblivious reconfigurable network designs (even with a randomized connection schedule) cannot achieve the same result as our semi-oblivious design: if the throughput guarantee must hold with probability 1, then the average latency must be strictly asymptotically greater for oblivious reconfigurable networks than for semi-oblivious ones.

1.1 Summary of Results and Techniques

In our abstraction of a reconfigurable network, a fixed set of N nodes communicates over a sequence of discrete time steps. In one time step, each node is allowed to send data to only one other node

Table 1: Bounds for reconfigurable networking with average latency constrained by $L = \tilde{O}(gN^{1/g})$.

Goal	Average hop-count	Throughput	Reference
Minimize network hops	g	—	naïve counting
Uniform multicommodity flow	g	$\frac{1}{g}$	[3]
Oblivious routing (w.h.p.)	$g+1$	$\frac{1}{g+1} - \delta \; \forall \delta > 0$	this work
Semi-oblivious routing (prob. 1)	$g+1-o(1)$	$\frac{1}{g+1} - \delta \; \forall \delta > 0$	this work
Oblivious routing (prob. 1)	$2g$	$\frac{1}{2g}$	[3] (uses VLB)

and to receive data from only one[1] other node. This time-varying connectivity pattern, called the *connection schedule*, may be randomized, but it must be predetermined in a demand-oblivious manner. To route messages through the network, nodes may forward data over links when they are available in the connection schedule, and they may buffer messages when the next link of the designated routing path is not yet available. The choice of routing paths is called the *routing scheme*. We allow data to be fractionally divided over routing paths (modeling the operation of randomly sampling one path per data packet) so the routing scheme is represented by specifying a fractional flow for each source-destination pair, at each time step. In an *oblivious* reconfigurable network this flow is predetermined, up to scaling, in a demand-oblivious manner. In a *semi-oblivious* reconfigurable network only the connection schedule is oblivious; the routing scheme may be demand-aware.

To place our results in context, it helps to reason a bit about the fundamental limits of communication in reconfigurable networks.

(1) **Throughput is bounded by the inverse of average hop-count.** A network design is said to have throughput r if it is able to serve any communication pattern whose ingress and egress rates, at each node in each time step, are bounded by r times the amount of data that may be transmitted on any link per time step. Adopting units in which link capacities equal 1, the total amount of demand originating in any time step is rN and the total link capacity is N. If the average routing path is composed of g network hops, then the rN units of demand originating in any time step will consume grN units of capacity on average, hence $gr \leq 1$. Guaranteeing throughput r therefore requires guaranteeing average hop-count at most $1/r$.

(2) **Hop-count g requires latency $L = \Omega(gN^{1/g})$.** A routing path originating at a given node is uniquely determined by the set of time steps at which the path traverses network hops. (This is because the connection schedule specifies a *unique* node that is allowed to receive messages from any given node at any given time.) Hence, in order for any node to be able to reach any other node within L time steps using a routing path of g or fewer hops, it must be the case that $\sum_{i=0}^{g} \binom{L}{i} \geq N$. The solution to this inequality is $L = \Omega(gN^{1/g})$. A more complicated counting argument, which we omit, establishes the same lower bound on *average* latency, even if the bound of g hops per path is relaxed to hold only on average.

[1]More generally one could impose a degree constraint, d, on the number of nodes to/from which one node can send/receive data in a single time step. See [3] for a more thurough explanation of this.

These considerations establish a sort of *speed-of-light barrier* for reconfigurable networking. Even without the constraint of obliviousness, delivering messages within $O(gN^{1/g})$ time steps on average requires g-hop paths, hence limits throughput to $1/g$. Oblivious or semi-oblivious network designs can thus be evaluated in relation to this benchmark. Table 1 presents a comparison of bounds for various reconfigurable networking goals, standardizing on an average latency constraint of $L = \tilde{O}(gN^{1/g})$ where g could be any positive integer (fixed, independent of N). As noted above, even if we ignore capacity constraints and connect all source-destination pairs using the minimum number of network hops subject to this latency constraint, average path length g is unavoidable. Optimal (demand-aware) routing schemes for the uniform multicommodity flow match this bound, whereas optimal oblivious routing schemes require average path length $2g$ [3]. The routing schemes presented in this paper have average path length $g+1$ (minus a $o(1)$ in the case of semi-oblivious routing), matching the "speed-of-light barrier" to within an additive 1. We also present lower bounds establishing that this result is the best possible.

The formal statement of our main results generalizes the foregoing discussion by allowing the target throughput rate to be any number (fixed, independent of N) in the interval $(0, \frac{1}{2}]$.

Theorem 1. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and let*

$$L_{upp}(r, N) = gN^{1/g} \tag{1}$$

$$L_{low}(r, N) = g\left((\varepsilon N)^{1/g} + N^{1/(g+1)}\right) \tag{2}$$

Assuming $\varepsilon \neq 1$:

(1) there exists a family of distributions over ORN designs for infinitely many network sizes N which attains maximum latency $\tilde{O}(L_{upp}(r, N))$, and achieves throughput r with high probability;

(2) for infinitely many network sizes, there exists a single, fixed ORN design that attains maximum latency $\tilde{O}(L_{upp}(r, N))$, and achieves throughput r with high probability over the uniform distribution on permutation demands;

(3) there exists a family of distributions over semi-oblivious reconfigurable network designs for infinitely many network sizes N which attains maximum latency $\tilde{O}(L_{upp}(r, N))$ with high probability (and in expectation) over time-stationary demands, and achieves throughput r with probability 1;

(4) furthermore, any fixed ORN design $\mathcal{R}$ of size N which achieves throughput r with high probability over time-stationary demands must suffer at least $\Omega(L_{low}(r, N))$ maximum latency.

The upper and lower bounds on lines (1)-(2) match to within a constant factor for most values of r: when $\frac{1}{r} \notin \bigcup_{m=2}^{\infty} \left(m - \frac{2}{2^m}, m\right]$ then $\varepsilon \geq 2^{-g}$, so $L_{low} \geq \frac{1}{2} L_{upp}$. The latency of our reconfigurable network designs is $L_{upp} \cdot \tilde{O}(\log N)$, hence the upper and lower bounds in Theorem 1 agree within a $\tilde{O}(\log N)$ factor for most values of r. See Figure 1 for a visualization of these bounds. Additionally, like in [40] we condition against $\varepsilon = 1$. This is due to requiring a strictly positive slack factor between the throughput r and $\frac{1}{g+1}$.

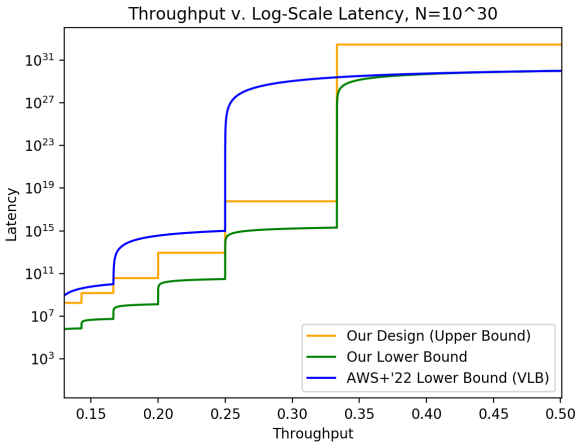

Figure 1: Throughput versus log-scale maximum latency. Tradeoff curves $\tilde{O}(L_{upp})$ and L_{low}, when compared against the lower bound of [3], on an ORN containing 10^{30} nodes. Whenever throughput is less than $\frac{1}{3}$, our design beats VLB.

We conclude this section by sketching how our routing scheme differs from VLB, and how we analyze it to obtain the bounds stated above. Both schemes construct routing paths composed of *spraying hops*, which transport messages from the source to a random intermediate node, and *direct hops*, which deliver messages from the intermediate node to the destination.

In both cases the analysis of the routing scheme entails showing that the spraying hops and the direct hops distribute load evenly over the network links, whenever the routing scheme is used to serve a *permutation demand*: a communication pattern where each source node s seeks to communicate at rate r with a single destination $\sigma(s)$, and the function σ is a permutation of $[N]$. For VLB this is easy: intermediate nodes are sampled uniformly at random, so the distribution of (source, intermediate node) pairs and the distribution of (intermediate node, destination) pairs are both uniform over the set of all pairs of nodes in the network; a symmetry argument then suffices to conclude that both the spraying hops and the direct hops distribute load evenly over all links.

In our routing scheme, routing paths consist of just one spraying hop followed by a direct path to the destination. Thus, the intermediate node must be either the source itself, or one of the nodes reachable by a direct link from the source node during the first L time steps after the message originates. For $L < N-1$ it is impossible for the intermediate node to be uniformly distributed, conditional on the source. Consequently (intermediate node, destination) pairs in our routing scheme are also not uniformly distributed. This non-uniform distribution retains some dependence on the permutation σ that associates sources with destinations. Hence it is unclear how to guarantee that for every permutation σ, flow traveling on direct hops will be uniformly distributed over the edges of the network.

Our main innovation lies in the way we construct a connection schedule and routing scheme to ensure (approximately) uniform distribution of load over edges. The use of a single spraying hop inevitably reduces the amount of randomness in the conditional distribution of the intermediate node given the source, and we must find a way to regain the lost randomness without adding extra spraying hops. To do so, we exploit a novel source of randomness: we randomize the *timing* of the direct hops. Prior work [3] used a connection schedule based on identifying the node set $[N]$ with a vector space over a finite field, and associating time steps with scalar multiples of the elementary basis vectors. To each pair of nodes one could then associate a direct path corresponding to the (unique) representation of the difference of the node identifiers as a linear combination of elementary basis vectors. Thus, the timing of direct hops was uniquely determined, given the location of the intermediate node.

In our connection schedule we again identify $[N]$ with a vector space over a finite field. However, there are two key differences. First, in some of our designs, the identification of $[N]$ with a finite vector space is done using a uniformly random one-to-one correspondence. This allows us to reduce the analysis of our (randomized) connection schedule to average-case analysis of a fixed connection schedule, when the demand matrices are conjugated by a uniformly random permutation matrix. Second, and more importantly, rather than defining the connection schedule using a basis of this vector space, we use an overcomplete system of vectors which we call a *constellation*. Constellations in a g-dimensional vector space have the property that every g-element subset forms a basis. (In other words, they represent the uniform matroid of rank g.) Our routing scheme constructs direct paths between two nodes by sampling a random g-element subset of the constellation, representing the difference between the nodes' identifiers as a linear combination of those g vectors, and using the corresponding g time steps of the connection schedule to form the direct path.

To show that this method distributes load approximately uniformly over edges, we decompose the load on any given edge as a sum of $g+1$ random variables, each of which can be interpreted as a bilinear form evaluated on a pair of vectors representing the number of paths from each source node to the tail of the given edge, and from the head of the given edge to each destination node. The pair of vectors is sampled at random from an orbit of the permutation group S_N, which acts on pairs of vectors either by permuting the coordinates of one of them (in the case when we're analyzing a uniformly random permutation demand) or by permuting the coordinates of both simultaneously (in the case when we're identifying the node set with a vector space using a random bijection). In both cases, we prove an exponential tail bound for the value of the bilinear form on a vector pair randomly sampled from the permutation-group orbit. When the permutation acts on only one element of the ordered pair, the relevant exponential tail bound follows easily from the Chernoff bound for negatively associated

random variables [13]. When the permutation acts on both vectors simultaneously, the negative association property does not hold and we take a more indirect approach, using a 3-coloring of the node set $[N]$ to decompose the bilinear form into three parts, each of which can be shown to satisfy an exponential tail bound after a suitable conditioning. We believe the resulting exponential tail bound for bilinear forms may be of independent interest.

To improve the high-probability bound on throughput to a bound that holds with probability 1, we adopt a semi-oblivious routing scheme that is a hybrid of a *primary scheme* identical to the oblivious scheme sketched above, and a *failover scheme* which is also oblivious, to be used in the (low-probability) case that the primary scheme produces an infeasible flow. The failover scheme has latency $\tilde{O}(N)$ and resembles VLB, distributing flow over two-hop paths from the source to the destination by routing through an intermediate node sampled from a nearly-uniform distribution. The challenge is to modify the connection schedule to ensure that enough two-hop paths exist between every source and destination. We accomplish this by using a time-varying constellation in place of the fixed constellation used by the routing scheme sketched above. The time-varying sequence of constellations that we construct forms a sort of combinatorial design, covering every vector with non-zero coordinates an equal number of times. This equal-coverage property is the key to proving the that the failover routing scheme balances load evenly. For further details on this Semi-Oblivious Reconfigurable Network Design, please see the full version of our paper.

Our lower bound. Our lower bound proof is heavily inspired by the lower bound proof of [3], thus we leave the proof of Theorem 1.4 to the full version of our paper. We build a family of $N!$ linear programs, one for each permutation on the node set, that each maximize throughput subject to a maximum latency constraint L. We then take the dual, find a good dual solution, and analyze the objective value of each dual solution. We then bound the expected objective value across the whole set, and use this to bound the achievable throughput with high probability. Interestingly, this lower bound result also applies to the guaranteed throughput rate of semi-oblivious designs – where the connection schedule must be pre-committed to, but the routing algorithm may be adaptive with respect to traffic.

1.2 Related Work

The most important related works, [3, 40], are summarized above in Section 1.

Oblivious routing in general networks. Extensive theoretical work in oblivious routing considers the competitive ratio in congestion achievable in general networks, when compared to an adaptive optimal routing. [33] proved the existence of a polylog n-competitive algorithm for this problem, the competitive ratio later improved upon by [24]. [5, 9, 24] then developed poly-time algorithms to achieve this result. Later, these algorithms were implemented and tested in wide-area networks [4]. [34] further improved to a $\log n$-competitive oblivious routing scheme, based on multiplicative weights and FRT's randomized approximation of general metric spaces by tree metrics [14]. This improved algorithm was again demonstrated in wide-area networks by [27].

Some works add additional constraints to this problem. For example, [21] found a polylog n-competitive routing scheme oblivious to both traffic and the cost functions of edges, and [16] finds a polylog n-competitive ratio when constraining the number of physical hops in paths that both the oblivious routing scheme, and the adaptive benchmark, can use. They also give an algorithm to achieve this. These works assume a fixed graph topology, while our work aims to co-design a network topology and routing scheme. They also examine congestion, a related but not analogous measure to our definition of throughput, make a guaranteed bound on that congestion instead of a probabilistic bound, and (with the exception of [16]) make little attempt to bound latency.

Randomized Oblivious Routing. There is also extensive work focused on oblivious routing with randomness. This problem is often focused on packet routing, and aims to obliviously choose a single path to route traffic on. It is well known that any such deterministic oblivious routing on a graph of degree d suffers $\Omega(\sqrt{N}/d)$ congestion from an adversarial permutation demand. [10, 25]. Valiant tackles this problem with Valiant Load Balancing, a randomized technique which gives a $\log n$-expected congestion bound on the d-dimensional hypercube, butterfly, and mesh networks [37, 38]. He later provided a lower bound in these contexts [36]. A similar procedure is used in ROMM routing in the hypercube, which selects a larger number of intermediate nodes within the sub-cube containing both the source and destination, and trades off load balancing with latency [30, 31]. These works differ from ours in that they aim to route discretized packets on paths, and look at the congestion that occurs from worst-case traffic.

[2] showed that in bit-serial routing, any random oblivious algorithm on a polylog degree network requires $O(\log^2 n/\log\log n)$ bit-steps with high probability for almost all permutation traffic, assuming $\log n$-bit messages, extending the Borodin-Hopcroft bound for deterministic algorithms. [27] examines a partially adaptive (or, semi-oblivious) routing, in which the router precommits to a set of $\log N$ paths between each pair of vertices, and at runtime may only send flow on one of the precommitted paths. This approach was later shown to be polylog n-competitive by [43]. Since oblivious routing under the same sparsity constraint cannot be polylog n-competitive, this constitutes an asymptotic separation between the power of semi-oblivious and oblivious routing. To the best of our knowledge [43] constitutes the first provable asymptotic separation between semi-oblivious and oblivious routing in the literature, and the separation that we prove (in the full version of our paper) is the second such result.

A work that closely models the problem we ask [22], gives a $O(log^2 n)$-competitive algorithm with high probability over random demands in directed graphs, and showed that one cannot do better than $O(\log n/\log\log n)$-competitive with any constant probability. Like in non-randomized oblivious routing, they also assume a fixed graph topology, and do not attempt to bound latency.

ORN Proposals. Although [3] is first to name the ORN paradigm, it was used earlier in proposed network architectures and designs. Rotornet [18] and Sirius [7] both use optical circuit switches to build a reconfigurable fabric, and Shoal [35] uses electronic circuit switches. These works demonstrate different ways to implement

ORNs using physical hardware, however they all use similar connection and routing schedules that maximize throughput, at the expense of latency. Opera [29] combines the ORN paradigm with lengthened time slots, high node degrees, and some adaptive routing. This allows a separation into two traffic classes, low-latency and throughput-sensitive. However the design makes significant assumptions about the traffic workload, limiting its flexibility. Cerberus [20] uses a modification of Rotornet as one component of an optical datacenter network, along with demand-aware reconfiguration and static graphs.

[1] used the degree of the time-collapsed connection schedule, or *emulated graph*, of an ORN design to bound its throughput, latency, and buffer requirement. Using this, the authors derived a formula for the ideal degree d to use for the emulated graph in order to maximize throughput in a buffer-constrained network. The authors proposed MARS, an ORN design that emulates a de Bruijn graph with this ideal degree, to achieve near-optimal throughput under buffer constraints, and evaluated this design through simulation.

2 DEFINITIONS

Definition 1. A *connection schedule* of N nodes and period length T is a sequence of permutations $\boldsymbol{\pi} = \pi_0, \pi_1, \ldots, \pi_{T-1}$, each mapping $[N]$ to $[N]$. $\pi_k(i) = j$ means that node i is allowed to send one unit of flow to node j during any timestep t such that $t \equiv k \pmod{T}$.

The *virtual topology* of the connection schedule $\boldsymbol{\pi}$ is a directed graph $G_{\boldsymbol{\pi}}$ with vertex set $[N] \times \mathbb{Z}$. The edge set of $G_{\boldsymbol{\pi}}$ is the union of two sets of edges, E_{virt} and E_{phys}. E_{virt} is the set of *virtual edges*, which are of the form $(i, t) \to (i, t+1)$ and represent flow waiting at node i during the timestep t. E_{phys} is the set of *physical edges*, which are of the form $(i, t) \to (\pi_t(i), t+1)$, and represent flow being transmitted from i to $\pi_t(i)$ during timestep t.

We interpret a path in $G_{\boldsymbol{\pi}}$ from (a, t) to (b, t') as a potential way to transmit one unit of flow from node a to node b, beginning at timestep t and ending at some timestep $t' > t$. Let $\mathcal{P}(a, b, t)$ denote the set of paths in $G_{\boldsymbol{\pi}}$ starting at the vertex (a, t) and ending at some (b, t') for any $t' > t$, and let $\mathcal{P}_L(a, b, t)$ be the set of such paths for which $t' - t \le L$. Finally, let $\mathcal{P} = \bigcup_{a,b,t} \mathcal{P}(a, b, t)$ denote the set of all paths in $G_{\boldsymbol{\pi}}$.

Definition 2. A *flow* is a function $f : \mathcal{P} \to [0, \infty)$. For a given flow f, the amount of flow traversing an edge e is defined as:

$$F(f, e) = \sum_{P \in \mathcal{P}} f(P) \cdot \mathbf{1}_{e \in P}$$

We say that f is *feasible* if for every physical edge $e \in E_{\text{phys}}$, $F(f, e) \le 1$. Note that in our definition of feasible, we allow virtual edges to have unlimited capacity.

Definition 3. An *oblivious routing scheme* R is a set of functions $R(a, b, t) : \mathcal{P} \to [0, 1]$, one for every tuple $(a, b, t) \in [N] \times [N] \times \mathbb{Z}$, such that:

(1) For all $(a, b, t) \in [N] \times [N] \times \mathbb{Z}$, $R(a, b, t)$ is a probability distribution supported on $\mathcal{P}(a, b, t)$.
(2) R has period T. In other words, $R(a, b, t)$ is equivalent to $R(a, b, t+T)$ (except with all paths transposed by T timesteps).

Definition 4. An *Oblivious Reconfigurable Network (ORN) design* $\mathcal{R}$ consists of both a connection schedule π_k and an oblivious routing scheme R.

Definition 5. A *demand-aware routing scheme* $\{S_\sigma : \sigma \text{ permut on } [N]\}$ is a set of functions $S_\sigma(a, t) : \mathcal{P} \to [0, 1]$, one for every tuple $(a, t) \in [N] \times \mathbb{Z}$ and permutation σ on $[N]$, such that:

(1) for all $(a, t, \sigma) \in [N] \times \mathbb{Z} \times S_N$, $S_\sigma(a, t)$ is a probability distribution supported on $\mathcal{P}(a, \sigma(a), t)$.
(2) S_σ has period T. In other words, $S_\sigma(a, t)$ is equivalent to $S_\sigma(a, t+T)$ (except with all paths transposed by T timesteps).

Definition 6. A *Semi-Oblivious Reconfigurable Network (SORN) Design* $\mathcal{S}$ consists of a connection schedule π_k and a demand-aware routing scheme $\{S_\sigma : \sigma \text{ permut on } [N]\}$.

Definition 7. The *latency* $L(P)$ of a path P in $G_{\boldsymbol{\pi}}$ is equal to the number of edges it contains (both virtual and physical). Traversing any edge in the virtual topology (either virtual or physical) is equivalent to advancing in time by one timestep, so the number of edges in a path equals the elapsed time. For an ORN Design $\mathcal{R}$ or SORN design $\mathcal{S}$, the *maximum latency* is the maximum over all paths P which may route flow.

$$L_{max}(\mathcal{R}) = \max_{P \in \mathcal{P}} \{L(P) : \exists a, b, t \text{ for which } R(a, b, t, P) > 0\}$$
$$L_{max}(\mathcal{S}) = \max_{P \in \mathcal{P}} \{L(P) : \exists a, t, \sigma \text{ for which } S_\sigma(a, t, P) > 0\}$$

The *average (or normalized) latency* is the weighted average across all possible demand pairs and all paths P which may route flow.

$$L_{avg}(\mathcal{R}) = \frac{1}{N^2 T} \sum_{a,b,t} \sum_{P \in \mathcal{P}(a,b,t)} R(a, b, t, P) L(P)$$
$$L_{avg}(\mathcal{S}) = \frac{1}{NTN!} \sum_{\sigma,a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t)} S_\sigma(a, t, P) L(P)$$

Definition 8. A *demand matrix* is an $N \times N$ matrix which associates to each ordered pair (a, b) a rate of flow to be sent from a to b. A *demand function* D is a function that associates to every $t \in \mathbb{Z}$ a demand matrix $D(t)$ representing the amount of flow $D(t, a, b)$ originating between each source-destination pair at timestep t.

A *permutation demand* D_σ is a demand function in which every demand matrix is the permutation matrix defined by $\sigma : [N] \to [N]$.

Definition 9. If R is an oblivious routing scheme and D is a demand function, the *induced flow* $f(R, D)$ is defined by:

$$f(R, D) = \sum_{(a,b,t) \in [N] \times [N] \times \mathbb{Z}} D(t, a, b) R(a, b, t).$$

If $\{S_\sigma : \sigma \text{ permut on } [N]\}$ is a demand-aware routing scheme and D_σ is a permutation demand function (possibly scaled by some constant), then the induced flow is defined by $f(S_\sigma, D_\sigma)$.

Definition 10. An ORN Design $\mathcal{R}$ *guarantees throughput* r if the induced flow $f(R, rD)$ is feasible whenever for all t, the row and column sums of $D(t)$ are bounded above by 1. (Such matrices $D(t)$ are called *doubly sub-stochastic*.) An ORN Design $\mathcal{R}$ guarantees throughput r *with respect to time-stationary demands* if for every time-stationary demand function D with row and column sums

bounded by 1, then the induced flow $f(R, rD)$ is feasible. An easy application of the Birkhoff-von Neumann Theorem establishes the following: in order for an ORN design to guarantee throughput r with respect to time-stationary demands, it is necessary and sufficient that it guarantee throughput r with respect to permutation demands.

An SORN design $\mathcal{S}$ *guarantees throughput* r (with respect to permutation demands) if, for every permutation demand D_σ, the induced flow $f(S_\sigma, rD_\sigma)$ is feasible for all t.

Definition 11. A distribution over ORN designs $\mathscr{R}$, is said to *achieve throughput r with high probability* if, for any $d \geq 1$ and demand function D such that $D(t)$ is doubly sub-stochastic for all t, routing rD on a random $\mathcal{R} \sim \mathscr{R}$ induces a feasible flow with probability at least $1 - C_d/N^d$, where C_d is a constant that may depend on d.

Similarly, $\mathscr{R}$ is said to *achieve throughput r with high probability under the uniform distribution on permutation demands* if, for uniformly random permutations σ and any $d \geq 1$, the induced flow $f(R, rD_\sigma)$ is feasible with probability at least $1 - C_d/N^d$, where C_d is a constant that may depend on d, and the randomness is over both the draw of $\mathcal{R}$ from $\mathscr{R}$ and the draw of σ from the uniform distribution over permutations. In the special case when $\mathscr{R}$ is a point-mass distribution on a singleton set $\{\mathcal{R}\}$, we say that the fixed design $\mathcal{R}$ achieves throughput r with high probability under the uniform distribution over permutation demands.

Definition 12. A distribution over SORN designs $\mathscr{S}$, is said to *achieve maximum latency L with high probability under the uniform permutation distribution* if, over uniformly random permutation σ and for any $d \geq 1$, routing rD_σ on a random $\mathcal{S} \sim \mathscr{S}$ uses paths of maximum latency L with probability at least $1 - C_d/N^d$, where C_d is a constant that may depend on d. In the special case when $\mathscr{S}$ is a point-mass distribution on a singleton set $\{\mathcal{S}\}$, we say that the fixed design $\mathcal{S}$ achieves maximum latency L with high probability under the uniform distribution over permutation demands.

An SORN design $\mathcal{S}$ *achieves maximum latency L with high probability* (under the uniform permutation distribution) if, for uniformly random permutations σ and any $d \geq 1$, $L_{max}(S_\sigma) = \max_{P \in \mathcal{P}}\{L(P) : \exists a, t \text{ for which } S_\sigma(a, t, P) > 0\}$ is no more than L with probability at least $1 - C_d/N^d$, where C_d is a constant that may depend on d.

Similarly, a distribution over SORN designs $\mathscr{S}$ achieves maximum latency L with high probability if, for any $d \geq 1$ and fixed permutation demand D_σ, routing D_σ on a random $\mathcal{S} \sim \mathscr{S}$ sends flow on paths on latency no more than L with probability at least $1 - C_d/N^d$, where C_d is a constant that may depend on d.

Definition 13. A *round robin* for a group of nodes S of size k, $\{s_0, \ldots, s_{k-1}\}$ is a schedule of $k-1$ timesteps in which each element of S has a chance to send directly to each other element exactly once; during timestep $t \in [k-1]$ node s_i may send to $s_{i+t \mod k}$.

3 UPPER BOUND: OBLIVIOUS DESIGN

In this section we prove Theorem 1, parts 1 and 2, restated below.

Theorem 1.1-1.2. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$, and let $L_{upp}(r, N)$ be the function*

$$L_{upp}(r, N) = gN^{1/g}$$

Then assuming $\frac{1}{r} \notin \mathbb{Z}$, there exists a family of distributions over ORN designs for infinitely many network sizes N which attains maximum latency $\tilde{O}(L_{upp}(r, N))$, and achieves throughput r with high probability. Furthermore, under the same assumption on r, for infinitely many network sizes there exists a fixed distribution over ORN designs which attains maximum latency $\tilde{O}(L_{upp}(r, N))$, and achieves throughput r with high probability under the uniform distribution.

We will begin by constructing an ORN design $\mathcal{R}^0$ which is parameterized by N, g, and C, where C is a parameter which we set during our analysis to a suitable function of N and r designed to achieve the appropriate tradeoffs between throughput and latency. We will then analyze $\mathscr{R}_N(g, C)$, a distribution over all ORN designs $\mathcal{R}^\tau$ which are equivalent to $\mathcal{R}^0$ up to re-labeling of nodes, and show that it satisfies the conclusion of Theorem 1.1. Furthermore, we will show that the fixed design $\mathcal{R}^0$ itself satisfies the conclusion of Theorem 1.2. We make use of the following definition of a *constellation* in our design.

Definition 14. A (C, g)-*constellation* in $\mathbb{F}_p^g$ is a sequence of $C(g+1)$ vectors for which the following property holds. Any set of g distinct vectors forms a basis over the vector space $\mathbb{F}_p^g$.

3.1 Connection Schedule

The connection schedule of $\mathcal{R}^0$, like the Vandermonde Basis Scheme of [3], is based on round-robin phases (cf. Definition 13) defined by Vandermonde vectors. However, in the case of $\mathcal{R}^0$, these vectors will form a (C, g)-constellation. We interpret the set of nodes as elements of the vector space $\mathbb{F}_p^g$ over the prime field $\mathbb{F}_p$, where $N = p^g$. Each node $a \in [N]$ can then be interpreted as a unique g-tuple $(a_1, a_2, \ldots, a_g) \in \mathbb{F}_p^g$.

During this connection schedule, each node will participate in a series of round robins, each defined by a single Vandermonde vector of the form $\boldsymbol{v}(x) = (1, x, x^2, \ldots, x^{g-1})$. The period length of the connection schedule is $T = C(g+1)(p-1)$, and one full period of the schedule consists of $C(g+1)$ consecutive round robins called *Vandermonde phases* or simply *phases*, each of length $(p-1)$ timesteps. The $C(g+1)$ phases constituting one period of the schedule are defined by distinct Vandermonde vectors of the form $\boldsymbol{v}(x) = (1, x, \ldots, x^{g-1})$. No property of the Vandermonde vectors other than distinctness is required – any set of $C(g+1)$ distinct Vandermonde vectors forms a (C, g)-constellation as desired. Since Vandermonde vectors are parameterized by elements $x \in \mathbb{F}_p$, we require $p \geq C(g+1)$ to ensure that sufficiently many distinct Vandermonde vectors exist. The set of Vandermonde phases in the (C, g)-constellation will be grouped into $(g+1)$ non-overlapping *phase blocks*, each phase block consisting of C phases.

More formally, we identify each congruence class $k \pmod T$ with a phase number x and a scale factor s, $0 \leq x < p$ and $1 \leq s < p$, such that $k = (p-1)x + s - 1$. It is useful to think of timesteps as being indexed by ordered pairs (x, s) rather than by the corresponding congruence class mod T, so we will sometimes abuse notation and refer to timestep (x, s) in the sequel, when we mean $k = (p-1)x + s - 1$. The connection schedule of $\mathcal{R}^0$, during timesteps $t \equiv k \pmod T$, uses permutation $\pi_k^0(a) = a + s\boldsymbol{v}(x)$, where x and s are the phase number and scale associated to k. Thus, each phase takes $(p-1)$ timesteps, and allows each node a

to connect with nodes a' where the difference $a' - a$ belongs to the one-dimensional linear subspace generated by $\boldsymbol{v}(x)$.

As described above, $\mathscr{R}_N(g, C)$ is a distribution over all ORN designs $\mathcal{R}^\tau$ which are equivalent to $\mathcal{R}^0$ up to re-labeling. When we sample a random design $\mathcal{R}^\tau$, we sample a uniformly random permutation of the node set $\tau : \mathbb{F}_p^h \to \mathbb{F}_p^g$, producing the schedule $\pi_k^\tau(a) = \tau^{-1}\left(\pi_k^0(\tau(a))\right)$. Note that, for every edge from node a to node $\pi_t^\tau(a)$ in $\mathcal{R}^\tau$, there is a unique equivalent edge from $\tau(a)$ to $\tau(\pi_t^\tau(a))$ in $\mathcal{R}^0$.

3.2 Routing Scheme

Our routing scheme for $\mathcal{R}^0$ constructs routing paths composed of at most one physical hop in each of $g+1$ consecutive phase blocks. Such a path can be identified by the node and timestep at which it originates, the phases in which it traverses a physical hop, and the scale factors applied to the Vandermonde vectors defining each of those phases. Our first definition specifies a structure called a *pseudo-path* that encodes all of this information.

Definition 15. A k-hop *pseudo-path* from a to b starting at time t is a sequence of ordered pairs $(x_1, \alpha_1), \ldots, (x_k, \alpha_k)$ such that:

- $x_1, \ldots, x_k$ are phases belonging to distinct, consecutive phase blocks beginning with the first complete phase block after time t;
- $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_p$ are scalars;
- $b - a = \alpha_1 \boldsymbol{v}(x_1) + \alpha_2 \boldsymbol{v}(x_2) + \cdots + \alpha_k \boldsymbol{v}(x_k)$.

A *non-degenerate* pseudo-path is one satisfying $\alpha_1 \neq 0$ and $\alpha_k \neq 0$.

The path corresponding to a pseudo-path is the path in the virtual topology that starts at a, traverses physical edges in timesteps $k_i = (x_i, \alpha_i)$ for all i such that $\alpha_i \neq 0$, and traverses virtual edges in all other timesteps.

Note that the path corresponding to a k-hop pseudo-path may contain fewer than k physical hops. Two distinct pseudo-paths may correspond to the same path, if the only difference between the pseudo-paths lies in the timing of the phases with $\alpha_j = 0$, i.e. the phases in which no physical hop is taken. Distinguishing between pseudo-paths that correspond to the same path is unnecessary for the purpose of describing the edge sets of routing paths, but it turns out to be essential for the purpose of defining and analyzing the *distribution* over routing paths employed by our routing schemes.

Our oblivious routing scheme for $\mathcal{R}^0$ divides flow among routing paths in proportion to a probability distribution over paths defined as follows. To sample routing path from a to b starting at time t, we sample a uniformly random non-degenerate $(g+1)$-hop pseudo-path from a to b that starts at time t. We then translate this pseudo-path into the corresponding path, and use that as a routing path from a to b. In other words, our oblivious routing scheme divides flow among paths in proportion to the number of corresponding non-degenerate $(g+1)$-hop pseudo-paths.

To analyze the oblivious routing scheme, or even to confirm that it is well-defined, it will help to prove a lower bound on the number of solutions to the equation

$$b - a = \alpha_1 \boldsymbol{v}(x_1) + \cdots + \alpha_{g+1} \boldsymbol{v}(x_{g+1}) \tag{3}$$

that satisfy $\alpha_1 \neq 0$, $\alpha_{g+1} \neq 0$. For any $i \in [g+1]$ and $\beta \in \mathbb{F}_p$, there is a unique solution to (3) with $\alpha_i = \beta$. This is because the equation

$$b - a - \beta \boldsymbol{v}(x_i) = \sum_{j \neq i} \alpha_j \boldsymbol{v}(x_j)$$

is a system of g linear equations in g unknowns, with an invertible coefficient matrix. (Here we have used the fact that the vectors $\boldsymbol{v}(x_j)$ are distinct Vandermonde vectors, hence linearly independent.) Hence, the total number of solutions of (3) is p, and there is exactly one solution with $\alpha_1 = 0$ and exactly one solution with $\alpha_{g+1} = 0$. The number of solutions with $\alpha_i \neq 0$ and $\alpha_{g+1} \neq 0$ is therefore either $p-2$ or $p-1$. Since there are C^{g+1} ways to choose the $g+1$ distinct phases $x_1, \ldots, x_{g+1}$, we conclude that the number of non-degenerate $(g+1)$-hop pseudo-paths from a to b starting at time t is between $(p-2)C^{g+1}$ and $(p-1)C^{g+1}$.

The routing scheme of $\mathcal{R}^\tau$, for general τ, is defined using the bijection between the edges of $\mathcal{R}^\tau$ and those of $\mathcal{R}^0$. For any path from node a to node b in $\mathcal{R}^\tau$ there is a unique equivalent path from $\tau(a)$ to $\tau(b)$ in $\mathcal{R}^0$. To route from a to b in $\mathcal{R}^\tau$, simply apply the inverse of this bijection to the probability distribution over routing paths from $\tau(a)$ to $\tau(b)$ in $\mathcal{R}^0$.

3.3 Latency-Throughput Tradeoff

It is clear that any design $\mathcal{R}^\tau \sim \mathscr{R}_N(g, C)$ will have maximum latency $C(g+2)(p-1) < C(g+2)N^{1/g}$. (The factor of $g+2$ reflects the fact that messages wait for the duration of at most one phase block, then use the following $g+1$ phase blocks to reach their destination.) Thus, we focus on proving the achieved throughput rate with high probability in this section. Parts 1 and 2 of the following theorem correspond to parts 2 and 1 of Theorem 1, respectively.

Theorem 2. *Given a fixed throughput value r, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and assume $\varepsilon \neq 1$. As N ranges over the set of prime powers p^g for primes p exceeding $\max\left\{C(g+1), 2 + \frac{2}{1-\varepsilon}\right\}$, let $\gamma = \ln\left(\frac{g-\varepsilon-2/(p-2)}{g-1}\right)$ and $C = \frac{\log\log N}{\gamma^2} \ln(N)$. Then:*

1. *the design $\mathcal{R}^0$ achieves throughput r with high probability under the uniform distribution,*
2. *the family of distributions $\mathcal{R}_N(g, C)$ achieves throughput r with high probability.*

Note that if $\varepsilon = 1$, i.e. if $\frac{1}{r} \in \mathbb{Z}$, then there are no primes p which exceed $2 + \frac{2}{1-\varepsilon}$, therefore we condition against $\varepsilon = 1$.

Both parts of the theorem will be proven by focusing on the congestion of physical edges in the design $\mathcal{R}^0$. For the first part, the focus on edges in $\mathcal{R}^0$ is obvious. For the second part, we make use of the isomorphism between $\mathcal{R}^\tau$ and $\mathcal{R}^0$. Rather than considering a fixed demand function D and random design $\mathcal{R}^\tau$, we may consider a fixed design $\mathcal{R}^0$ and random demand function $D^\tau(t) = P^{-1}D(t)P$ where P denotes the permutation matrix with $P_{i,\tau(i)} = 1$ for all i.

Now, focusing on any particular edge $e \in E_{\text{virt}}(\mathcal{R}^0)$, we bound the probability that e is overloaded by breaking down the (random) amount of flow traversing e as a sum, over $0 \leq q \leq g$, of the amount of flow that crosses e on the $(q+1)$-th hop of a routing path. We will describe how to interpret each of these random amounts of flow as the value of a bilinear form on a pair of vectors randomly sampled

from an orbit of a permutation group action. (The bilinear form is related to the demand function D, and the pair of vectors is related to the routing scheme.) We will then use a Chernoff-type bound for the values of bilinear forms on permutation group orbits, to bound the probability that the amount of $(q+1)$-th hop flow crossing e is larger than average. Finally we will impose a union bound to show the probability that any edge gets overloaded is extremely small.

Existing Chernoff-type bounds for negatively associated random variables are sufficient for the tail bound in the first part of the theorem, but not for the second part. Instead, we prove the following novel tail bound for the distribution of bilinear sums on orbits of a permutation group action.

Theorem 3. *Suppose $\boldsymbol{u}, \boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying*

$$\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \geq CN \tag{4}$$

for some $C \geq 1$. Let D be any N-by-N doubly stochastic matrix and consider the bilinear form

$$B(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \neq j} D_{ij} x_i y_j. \tag{5}$$

Let $M = 1$ if D is a permutation matrix, and $M = N^2$ otherwise. If P is a uniformly random N-by-N permutation matrix then:

(1) for any $\gamma > 0$,

$$\Pr\left(B(\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma} \frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq M e^{-\frac{1}{2}\gamma^2 C}; \tag{6}$$

(2) for any $\gamma > 0$,

$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma} \frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq 15 M e^{-\frac{1}{100}\gamma^2 C}. \tag{7}$$

The proof of Theorem 3 is left to the full version of our paper.

Proof. **(Of Theorem 2.)** We may assume without loss of generality that the demand matrix $D(t)$ is doubly stochastic for all t. For part 1 of the theorem this is because $D(t)$ is assumed to be a random permutation matrix. For part 2, it is because every non-negative matrix whose row and column sums are bounded above by 1 can be made into a doubly stochastic matrix by (weakly) increasing each of the matrix entries [3]. Modifying the demand function in this way cannot decrease the induced flow on any edge, so it cannot increase the probability that $f(R, rD)$ is feasible. Thus, we will assume for the remainder of the proof that $D(t)$ is doubly stochastic for all t.

Fix an edge e and $0 \leq q \leq g$, and consider the amount of flow traversing edge e traveling on paths where edge e occurs in the $(q+1)$-th phase block[2] of the flow path. We will denote this value as the *amount of $(q+1)$-th hop flow traversing edge e*.[3]

First we examine $q = 0$. First-hop flow traversing edge e originates at source node tail(e) during the phase block preceding the one to which e belongs. There are $C(p-1)$ time steps during that phase block, and r units of flow per time step originate at tail(e). Each unit of flow is divided evenly among a set of at least $(p-2)C^{g+1}$ pseudo-paths, at most C^g of which begin with edge e as their first hop. (After fixing the first hop and the destination of a $(g+1)$-hop pseudo-path, the rest of the path is uniquely determined by the g-tuple of phases $x_2, \ldots, x_{g+1}$.) Hence, of the $rC(p-1)$ units of flow that could traverse e as their first hop, the fraction that actually do traverse e as their first hop is at most $\frac{C^g}{(p-2)C^{g+1}}$. Consequently, the amount of first-hop flow on e is bounded above by $\frac{rC(p-1)\cdot C^g}{(p-2)C^{g+1}} = \left(\frac{p-1}{p-2}\right) r$. (Note that this is not a probabilistic statement; the upper bound on first-hop flow holds with probability 1.) A symmetric argument shows that the amount of last-hop flow on e is bounded above by $\left(\frac{p-1}{p-2}\right) r$ as well.

Now suppose $1 \leq q \leq g-1$, and let X_i be the random variable realizing the amount of $(q+1)$-th hop flow traversing edge e due to source node i. Clearly, the total amount of $(q+1)$-th hop flow traversing e will be $\sum_i X_i$. Let I denote the interval of timesteps constituting the q^{th} phase block before the phase block that contains edge e; recall that this means I is made up of $C(p-1)$ consecutive timesteps. Let

$$\overline{D}_{ij} = \frac{1}{rC(p-1)} \sum_{t \in I} D(t)_{ij}$$

denote the (normalized) rate of flow demanded by source-destination pair (i, j) during phase block I. The normalizing factor makes $\overline{D}$ into a doubly stochastic matrix. Let $\rho_q^-(i, e)$ denote the number of q-hop pseudo-paths from i to tail(e) with non-zero first coefficient, and let $\rho_{g-q}^+(e, j)$ denote the number of $(g-q)$-hop pseudo-paths from head(e) to j with non-zero last coefficient. Finally, let $\rho_{g+1}(i, j)$ denote the number of non-degenerate $(g+1)$-hop pseudo-paths from i to j. Of the flow that originates at i with destination j during time window I, the fraction of flow that traverses edge e under our routing scheme for $\mathcal{R}^0$ is $\rho_q^-(i, e) \cdot \rho_{g-q}^+(e, j)/\rho_{g+1}(i, j)$. Hence,

$$\begin{aligned} X_i &= \sum_{j \in [N],\, j \neq i} \frac{\rho_q^-(i, e) \cdot \rho_{g-q}^+(e, j)}{\rho_{g+1}(i, j)} \cdot \left(\sum_{t \in I} D(t)_{ij}\right) \\ &\leq \sum_{j \in [N],\, j \neq i} \frac{\rho_q^-(i, e) \cdot \rho_{g-q}^+(e, j) \cdot rC(p-1) \cdot \overline{D}_{ij}}{(p-2)C^{g+1}} \\ &= \left(\frac{p-1}{p-2}\right) r \sum_{j \in [N],\, j \neq i} \overline{D}_{ij} \left(\frac{\rho_q^-(i, e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e, j)}{C^{g-q}}\right) \end{aligned}$$

$$\sum_{i=1}^N X_i \leq \left(\frac{p-1}{p-2}\right) r \sum_{i \neq j} \overline{D}_{ij} \left(\frac{\rho_q^-(i, e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e, j)}{C^{g-q}}\right) = \sum_{i \neq j} \overline{D}_{ij} u_i v_j \tag{8}$$

where

$$u_i = \left(\frac{p-1}{p-2}\right) r \left(\frac{\rho_q^-(i, e)}{C^q}\right), \qquad v_j = \frac{\rho_{g-q}^+(e, j)}{C^{g-q}}. \tag{9}$$

To prove the first part of the theorem, Theorem 2.1, when the ORN design is fixed to be $\mathcal{R}^0$ and the demand function is the time-stationary demand D_σ for a random permutation σ, then

$$\sum_{i \neq j} \overline{D}_{ij} u_i v_j = \sum_{i \neq \sigma(i)} u_i v_{\sigma(i)} \leq \sum_{i=1}^N u_i v_{\sigma(i)}.$$

[2] We number phase blocks in a flow path using the convention that phase block 1 is the first *complete* phase block in the flow path. Recall from Section 3.2 that this is also the first phase block in which it is possible that the flow is transmitted on a physical edge.

[3] Note this is a different value than if edge e is the $(q+1)$-th physical hop traversed on the path. It may be the case that in some earlier phase blocks of the path, flow may not have traversed any physical hop. If this is confusing, revisit *pseudo-paths* in Section 3.2.

The distribution of σ is the same as the distribution of $\tau \circ \pi$ where π is an arbitrary (non-random) permutation without fixed points, and τ is a uniformly random permutation. Letting P denote the permutation matrix representing τ, the amount of $(q+1)^{\text{th}}$ hop flow on edge e is stochastically dominated by

$$\sum_{i=1}^{N} u_i v_{\tau(\pi(i))} = B_\pi(\boldsymbol{u}, P\boldsymbol{v})$$

where B_π denotes the bilinear form $B_\pi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{N} x_i y_{\pi(i)}$.

Similarly, to prove the second part of the theorem, Theorem 2.2, recall that we are drawing a random ORN design $\mathcal{R}^\tau$ from the distribution $\mathscr{R}_N(C, r)$, and that the induced $(q+1)$-th hop flow on the edge of $\mathcal{R}^\tau$ corresponding to e, under demand function D, is equal to the induced $(q+1)$-th hop flow on edge e under demand function $P^{-1}DP$. Again letting P denote the permutation matrix representing τ, this induced flow is bounded above by

$$\sum_{i \neq j} (P^{-1}\overline{D}P)_{ij} u_i v_j = \sum_{i \neq j} \overline{D}_{ij} u_{\tau(i)} v_{\tau(j)} = B(P\boldsymbol{u},\, P\boldsymbol{v})$$

where B is the bilinear form $B(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \neq j} \overline{D}_{ij} x_i y_j$.

Hence, we are in a position to prove tail bounds on the induced $(q+1)$-th hop flow on edge e, using the Chernoff-type bounds in Theorem 3, provided we can estimate the norms $\|\boldsymbol{u}\|_1$, $\|\boldsymbol{v}\|_1$, $\|\boldsymbol{u}\|_\infty$, $\|\boldsymbol{v}\|_\infty$. For $\|\boldsymbol{u}\|_1$ we have $\|\boldsymbol{u}\|_1 = \frac{p-1}{p-2} \cdot \frac{r}{C^q} \cdot \sum_{i=1}^{N} \rho_q^-(i, e)$. The sum on the right side can be calculated by realizing that it counts the total number of q-hop pseudo-paths with non-zero first coefficient that end at tail(e). There are C^q ways of choosing a q-tuple of phases from the q phase blocks preceding the phase block containing e, for each such choice there are $(p-1)p^{q-1}$ ways to choose a sequence of coefficients beginning with a non-zero value. Hence,

$$\|\boldsymbol{u}\|_1 = \frac{p-1}{p-2} \cdot \frac{r}{C^q} \cdot (p-1)p^{q-1}C^q = \frac{(p-1)^2}{p(p-2)} \cdot p^q \cdot r.$$

Similarly,

$$\|\boldsymbol{v}\|_1 = \frac{p-1}{p} \cdot p^{g-q}.$$

Now we turn to bounding $\|\boldsymbol{u}\|_\infty$, $\|\boldsymbol{v}\|_\infty$ from above, which is tantamount to bounding the number of q-hop pseudo-paths from i to tail(e) and $(g-q)$-hop pseudo-paths from head(e) to j, with non-zero first and last coefficients respectively. One such upper bound is easy to derive: for each of the C^q many ways of selecting one phase $\boldsymbol{x}_i$ from each of the q phase blocks preceding tail(e), there is at most one q-hop pseudo-path from i to tail(e) using that sequence of phases. This is because the existence of two distinct such pseudo-paths would imply that the vector tail$(e) - i$ could be represented in two distinct ways as a linear combination of vectors in the set $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q\}$, violating linear independence. For an analogous reason, $\rho_q^+(\text{head}(e), j) \leq C^{g-q}$.

However, if $q \leq g/2$ then there is a tighter upper bound: $\rho_q^-(i, \text{tail}(e)) \leq C^{q-1}$. To see why, first observe that any $2q$ of the $C(g+1)$ Vandermonde vectors used in the $g+1$ phase blocks preceding edge e must be linearly independent, since $2q \leq g$. If $(x_1, \alpha_1), \ldots, (x_q, \alpha_q)$ and $(x'_1, \alpha'_1), \ldots, (x'_q, \alpha'_q)$ are two pseudo-paths from i to tail(e) then

$$\{(x_i, \alpha_i) \mid \alpha_i \neq 0\} \;=\; \{(x'_j, \alpha'_j) \mid \alpha'_j \neq 0\},$$

as otherwise the vector tail$(e) - i$ could be represented in two inequivalent ways as a linear combination of elements of $\{x_1, x'_1, x_2, x'_2, \ldots, x_q, x'_q\}$, contradicting linear independence. Consequently, when $q \leq g/2$, two distinct q-hop pseudo-paths from i to tail(e) can only differ in the choice of phases x_i with $\alpha_i = 0$. In other words, every q-hop pseudo-path from i to tail(e) has the same coefficient sequence $\alpha_1, \alpha_2, \ldots, \alpha_q$, and in constructing the corresponding phase sequence we have only one choice of phase when $\alpha_i \neq 0$ and C choices when $\alpha_i = 0$. Furthermore, there is at least one value of i, namely $i = 1$, for which $\alpha_i \neq 0$. Consequently, $\rho_q^-(i, \text{tail}(e)) \leq C^{q-1}$ when $q \leq g/2$, as claimed. An analogous argument proves that $\rho_q^+(\text{head}(e), j) \leq C^{g-q-1}$ when $g - q \leq g/2$. For every q, at least one of $q, g-q$ is less than or equal to $g/2$, and hence

$$\begin{aligned}
&\rho_q^-(i, \text{tail}(e)) \cdot \rho_q^+(\text{head}(e), j) \\
&\qquad \leq \max\{C^{q-1} \cdot C^{g-q},\, C^q \cdot C^{g-q-1}\} = C^{g-1} \\
&\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty \leq \left(\frac{p-1}{p-2}\right) r \left(\frac{\rho_q^-(i, \text{tail}(e)) \cdot \rho_q^+(\text{head}(e), j)}{C^g}\right) \\
&\qquad \leq \left(\frac{p-1}{p-2}\right) \frac{r}{C} \\
&\left(\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty}\right) \geq \frac{\frac{(p-1)^3}{p^2(p-2)} \cdot p^g \cdot r}{\frac{p-1}{p-2} \cdot \frac{r}{C}} = \left(\frac{p-1}{p}\right)^2 CN \geq \frac{1}{2} CN
\end{aligned}$$

for $p \geq 5$. If we observe that $\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N} = \frac{(p-1)^3}{p^2(p-2)} r < r$, then we may use Theorem 3 to conclude that for any $\gamma > 0$,

$$\begin{aligned}
\Pr\left(B_\pi(\boldsymbol{u}, P\boldsymbol{v}) \geq e^\gamma r\right) &\leq\; N^2 e^{-\frac{1}{4}\gamma^2 C} \\
\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^\gamma r\right) &\leq 15N^2 e^{-\frac{1}{200}\gamma^2 C}.
\end{aligned}$$

Supposing $C \geq \frac{\log\log N}{\gamma^2} \ln(N)$ for some positive integer, then we union bound over all $C(p-1)(g+1)N$ edges of the virtual topology and all $1 \leq q \leq g-1$ to find

$$\begin{aligned}
&\Pr[\text{any edge has} \geq e^\gamma r \ (q+1)\text{-th hop flow for any } 1 \leq q \leq g-1] \\
&\quad \leq NC(p-1)(g+1)(g-1) \cdot 15N^2 \left(e^{-\frac{1}{200}\gamma^2}\right)^C \\
&\quad \leq N^{3+1/g} \frac{\log\log N}{\gamma^2} \ln(N)(g^2-1) e^{-\frac{1}{200}\log\log N \ln(N)} \\
&\quad \leq \left(N^{3+1/g} \frac{\log\log N \ln(N)}{\gamma^2}(g^2-1)\right) N^{-\frac{1}{200}\log\log N} \\
&\quad \leq O\left(\frac{1}{\gamma^2 N^d}\right) \text{ for any constant } d.
\end{aligned}$$

This fulfills our definition of with high probability for fixed γ.

Finally, we need to show that if none of the bad events as described above occur, if every edge has at most $e^\gamma r$ $(q+1)$-th hop flow for $1 \leq q \leq g-1$, then no edge will be overloaded. Recall also that the $(q+1)$-th hop flow on e for $q \in \{0, g\}$ is $\left(\frac{p-1}{p-2}\right) r = r + \frac{r}{p-2}$. Recall also that $e^\gamma = \frac{g-\varepsilon-2/(p-2)}{g-1}$, $g = \lfloor \frac{1}{r} - 1 \rfloor$, and $\varepsilon = g + 1 - \left(\frac{1}{r} - 1\right) = 2 + g - \frac{1}{r}$. Hence, if no bad events occur, the induced flow on each edge will be bounded above by

$$2r + \frac{2r}{p-2} + (g-1)e^{\gamma} r = \left(2 + \frac{2}{p-2} + g - \varepsilon - \frac{2}{p-2}\right) r$$
$$= (2 + g - \varepsilon)\, r = \left(\frac{1}{r}\right) r = 1.$$

□

4 CONCLUSION AND OPEN QUESTIONS

In this paper, we showed that, compared to the guaranteed throughput versus latency tradeoff achieved in [3], a strictly superior latency-throughput tradeoff is achievable when the throughput bound is relaxed to hold with high probability. We showed that the same improved tradeoff is also achievable with guaranteed throughput under time-stationary demands, provided the latency bound is relaxed to hold with high probability and that the network is allowed to be semi-oblivious, using an oblivious (randomized) connection schedule but demand-aware routing. We proved that the latter result is not achievable by any fully-oblivious reconfigurable network design, marking a rare case in which semi-oblivious routing has a provable asymptotic advantage over oblivious routing.

Removing the logarithmic gap and when ε is small. Our designs only attain maximum latency $O(L_{upp}(r, N))$ up to a $\tilde{O}(\log N)$ factor, leaving a logarithmic gap between our upper and lower bounds. Is there an ORN or SORN design that achieves maximum latency $O(L_{upp}(r, N))$? Alternatively, is there a stronger lower bound than the one we presented in Theorem 1.4?

Additionally, when $\varepsilon^{1/g}$ is sub-constant, then $L_{upp}(r, N) > O(L_{low}(r, N))$. This leaves us with a small but measurable fraction of throughput values for which we cannot find ORN and SORN designs which achieve provably optimal throughput-latency tradeoffs, even up to a logarithmic factor. [3] handled this case by developing a second ORN family which sent flow on both h- and $(h+1)$-hop semi-paths. We believe a similar result for ORNs achieving throughput with high probability, and for SORNs, may be proven by considering larger numbers of constellations when routing the hop-efficient paths. However, we leave that to future work.

Time-varying demands. In order to prove our throughput-latency tradeoffs for SORN designs, we were required to restrict ourselves to time-stationary (permutation) demands. While this still shows that semi-oblivious routing has a provable asymptotic advantage over oblivious routing in the case of reconfigurable networks, it is desirable to find SORN designs which can handle time-varying demands. Our SORN design $\mathcal{S}_N(g, C)$ works for almost all time-varying demands. However, in the case that it must route all flow (from every starting timestep t) along 2-hop paths, there is no obvious way to "ramp back up" to sending flow on $(g+1)$-hop paths again without waiting for most flow in the network to clear, which would require almost 2 full periods, or iterations of the schedule.

Bridging the gap between theory and practice. As with previous work, we make several assumptions that do not hold in practice in order to make the analysis tractable. In particular, our model of ORNs does not account for propagation delay between nodes. In practice, it takes time for each message to traverse each physical link. Our model of ORNs can easily be adjusted to take this into account with our definition of the virtual topology, and the design itself could be modified by taking advantage of the fact that flow paths always take at most one physical hop per phase block. However, large propagation delays penalize solutions which take more physical hops, which inherently changes the attainable throughput versus latency tradeoffs in a real system. Once propagation delays become superlinear in N, one should always maximize throughput, since latency becomes dominated by propagation delay. It is worth exploring where and how this shift from a full tradeoff curve to a single optimal point occurs, as propagation delay increases.

Additionally, we assume fractional flow: each unit of flow can be fractionally divided and sent across multiple different paths. In a practical network, flow is sent in discrete packets, which cannot be divided. Due to this assumption, our model sends small fractions of flow from multiple paths across the same link. However in a real system, only one packet from one path may traverse the link during a single timestep. This may lead to queuing, which is best addressed using a congestion control system. Congestion control has a decades-long history of active research across various networking contexts. Our proposed designs present a new context for this area of research, and will likely require both adapting existing ideas from other contexts, as well as new innovations.

ACKNOWLEDGEMENTS

This work was supported in part by NSF grants CHS-1955125 and DBI-2019674, NSF Career Award 2239829, NSF Award 2331111, a Microsoft Investigator Fellowship, and research awards from Google and Cisco (23089533).

REFERENCES

[1] Vamsi Addanki, Chen Avin, and Stefan Schmid. 2023. Mars: Near-optimal throughput with shallow buffers in reconfigurable datacenter networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 7, 1 (2023), 1–43.

[2] William A. Aiello, F. T. Leighton, Bruce M. Maggs, and Mark Newman. 1991. Fast algorithms for bit-serial routing on a hypercube. *Mathematical systems theory* 24, 1 (1991), 253–271. https://doi.org/10.1007/BF02090402

[3] Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, Robert Kleinberg, and Rachit Agarwal. 2022. Optimal Oblivious Reconfigurable Networks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Rome, Italy) *(STOC 2022)*. Association for Computing Machinery, New York, NY, USA, 1339–1352. https://doi.org/10.1145/3519935.3520020

[4] David L. Applegate and Edith Cohen. 2003. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 25-29, 2003, Karlsruhe, Germany*, Anja Feldmann, Martina Zitterbart, Jon Crowcroft, and David Wetherall (Eds.). ACM, 313–324. https://doi.org/10.1145/863955.863991

[5] Yossi Azar, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Räcke. 2003. Optimal Oblivious Routing in Polynomial Time. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing* (San Diego, CA, USA) *(STOC '03)*. Association for Computing Machinery, New York, NY, USA, 383–388. https://doi.org/10.1145/780542.780599

[6] Moshe Babaioff and John Chuang. 2007. On the optimality and interconnection of valiant load-balancing networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 80–88.

[7] Hitesh Ballani, Paolo Costa, Raphael Behrendt, Daniel Cletheroe, Istvan Haller, Krzysztof Jozwik, Fotini Karinou, Sophie Lange, Kai Shi, Benn Thomsen, et al. 2020. Sirius: A Flat Datacenter Network with Nanosecond Optical Switching. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 782–797.

[8] Kashinath Basu, Ali Maqousi, and Frank Ball. 2020. Architecture of an end-to-end energy consumption model for a Cloud Data Center. In *2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*. IEEE, 1–6.

[9] Marcin Bienkowski, Miroslaw Korzeniowski, and Harald Räcke. 2003. A Practical Algorithm for Constructing Oblivious Routing Schemes. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures* (San

Diego, California, USA) *(SPAA '03)*. Association for Computing Machinery, New York, NY, USA, 24–33. https://doi.org/10.1145/777412.777418

[10] Allan Borodin and John E. Hopcroft. 1985. Routing, Merging, and Sorting on Parallel Models of Computation. *J. Comput. Syst. Sci.* 30 (1985), 130–145.

[11] Q. Cheng, A. Wonfor, J. L. Wei, R. V. Penty, and I. H. White. 2014. Demonstration of the feasibility of large-port-count optical switching using a hybrid Mach–Zehnder interferometer–semiconductor optical amplifier switch module in a recirculating loop. *Opt. Lett.* 39, 18 (Sep 2014), 5244–5247. https://doi.org/10.1364/OL.39.005244

[12] M. Ding, A. Wonfor, Q. Cheng, R. V. Penty, and I. H. White. 2017. Scalable, low-power-penalty nanosecond reconfigurable hybrid optical switches for data centre networks. In *2017 Conference on Lasers and Electro-Optics (CLEO)*. 1–2.

[13] Devdatt P Dubhashi and Desh Ranjan. 1996. Balls and bins: A study in negative dependence. *BRICS Report Series* 3, 25 (1996).

[14] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2004. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.* 69, 3 (2004), 485–497. https://doi.org/10.1016/j.jcss.2004.04.011

[15] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *Proceedings of ACM SIGCOMM*.

[16] Mohsen Ghaffari, Bernhard Haeupler, and Goran Zuzic. 2021. Hop-Constrained Oblivious Routing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy) *(STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 1208–1220. https://doi.org/10.1145/3406325.3451098

[17] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel Kilper. 2016. ProjecToR: Agile Reconfigurable Data Center Interconnect. In *Proceedings of the 2016 ACM SIGCOMM Conference* (Florianopolis, Brazil) *(SIGCOMM '16)*. Association for Computing Machinery, New York, NY, USA, 216–229. https://doi.org/10.1145/2934872.2934911

[18] Soudeh Ghorbani, Zibin Yang, P. Brighten Godfrey, Yashar Ganjali, and Amin Firoozshahian. 2017. DRILL: Micro Load Balancing for Low-Latency Data Center Networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 225–238. https://doi.org/10.1145/3098822.3098839

[19] Albert Greenberg, James Hamilton, David A Maltz, and Parveen Patel. 2008. The cost of a cloud: research problems in data center networks. , 68–73 pages.

[20] Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin. 2021. Cerberus: The power of choices in datacenter topology design-a throughput perspective. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 3 (2021), 1–33.

[21] Anupam Gupta, Mohammad Taghi Hajiaghayi, and Harald Räcke. 2006. Oblivious network design. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*. ACM Press, 970–979. http://dl.acm.org/citation.cfm?id=1109557.1109665

[22] MohammadTaghi Hajiaghayi, Jeong Han Kim, Tom Leighton, and Harald Räcke. 2005. Oblivious Routing in Directed Graphs with Random Demands. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing* (Baltimore, MD, USA) *(STOC '05)*. Association for Computing Machinery, New York, NY, USA, 193–201. https://doi.org/10.1145/1060590.1060619

[23] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. 2014. FireFly: A Reconfigurable Wireless Data Center Fabric Using Free-Space Optics. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (Chicago, Illinois, USA) *(SIGCOMM '14)*. Association for Computing Machinery, New York, NY, USA, 319–330. https://doi.org/10.1145/2619239.2626328

[24] Chris Harrelson, Kirsten Hildrum, and Satish Rao. 2003. A polynomial-time tree decomposition to minimize congestion. In *SPAA 2003: Proceedings of the Fifteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 7-9, 2003, San Diego, California, USA (part of FCRC 2003)*, Arnold L. Rosenberg and Friedhelm Meyer auf der Heide (Eds.). ACM, 34–43. https://doi.org/10.1145/777412.777419

[25] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. 1991. Tight Bounds for Oblivious Routing in the Hypercube. *Math. Syst. Theory* 24, 4 (1991), 223–232. https://doi.org/10.1007/BF02090400

[26] Isaac Keslassy, Cheng-Shang Chang, Nick McKeown, and Duan-Shin Lee. 2005. Optimal load-balancing. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, Vol. 3. IEEE, 1712–1722.

[27] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiunlin Lim, and Robert Soulé. 2018. Semi-Oblivious Traffic Engineering: The Road Not Taken. In *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, Sujata Banerjee and Srinivasan Seshan (Eds.). USENIX Association, 157–170. https://www.usenix.org/conference/nsdi18/presentation/kumar

[28] He Liu, Feng Lu, Alex Forencich, Rishi Kapoor, Malveeka Tewari, Geoffrey M. Voelker, George Papen, Alex C. Snoeren, and George Porter. 2014. Circuit Switching Under the Radar with REACToR. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. USENIX Association, Seattle, WA, 1–15. https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/liu_he

[29] William M. Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C. Snoeren, and George Porter. 2020. Expanding across time to deliver bandwidth efficiency and low latency. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. USENIX Association, Santa Clara, CA, 1–18. https://www.usenix.org/conference/nsdi20/presentation/mellette

[30] Ted Nesson and Lennart Johnsson. 1994. ROMM routing: A class of efficient Minimal routing algorithms. In *Parallel Computer Routing and Communication*, Kevin Bolding and Lawrence Snyder (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 185–199.

[31] Ted Nesson and S. Lennart Johnsson. 1995. ROMM Routing on Mesh and Torus Networks. In *Proceedings of the Seventh Annual ACM Symposium on Parallel Algorithms and Architectures* (Santa Barbara, California, USA) *(SPAA '95)*. Association for Computing Machinery, New York, NY, USA, 275–287. https://doi.org/10.1145/215399.215455

[32] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2013. Integrating Microsecond Circuit Switching into the Data Center. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 447–458. https://doi.org/10.1145/2486001.2486007

[33] H. Räcke. 2002. Minimizing congestion in general networks. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* 43–52. https://doi.org/10.1109/SFCS.2002.1181881

[34] Harald Räcke. 2008. Optimal Hierarchical Decompositions for Congestion Minimization in Networks *(STOC '08)*. Association for Computing Machinery, New York, NY, USA, 10 pages. https://doi.org/10.1145/1374376.1374415

[35] Vishal Shrivastav, Asaf Valadarsky, Hitesh Ballani, Paolo Costa, Ki Suh Lee, Han Wang, Rachit Agarwal, and Hakim Weatherspoon. 2019. Shoal: A Network Architecture for Disaggregated Racks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. USENIX Association, Boston, MA. https://www.usenix.org/conference/nsdi19/presentation/shrivastav

[36] Valiant. 1983. Optimality of a Two-Phase Strategy for Routing in Interconnection Networks. *IEEE Trans. Comput.* C-32, 9 (1983), 861–863. https://doi.org/10.1109/TC.1983.1676335

[37] Leslie G. Valiant. 1982. A Scheme for Fast Parallel Communication. *SIAM J. Comput.* 11, 2 (1982), 350–361. https://doi.org/10.1137/0211027

[38] Leslie G. Valiant and Gordon J. Brebner. 1981. Universal Schemes for Parallel Communication. (1981), 263–277. https://doi.org/10.1145/800076.802479

[39] Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T.S. Eugene Ng, Michael Kozuch, and Michael Ryan. 2010. C-Through: Part-Time Optics in Data Centers. In *Proceedings of the ACM SIGCOMM 2010 Conference* (New Delhi, India) *(SIGCOMM '10)*. Association for Computing Machinery, New York, NY, USA, 327–338. https://doi.org/10.1145/1851182.1851222

[40] Tegan Wilson, Daniel Amir, Vishal Shrivastav, Hakim Weatherspoon, and Robert Kleinberg. 2023. Extending Optimal Oblivious Reconfigurable Networks to All N. In *Proceedings of the SIAM Symposium on Algorithmic Principles of Computer Systems (APOCS)* (Florence, Italy).

[41] Rui Zhang-Shen and Nick McKeown. 2005. Designing a predictable internet backbone with valiant load-balancing. In *Quality of Service–IWQoS 2005: 13th International Workshop, IWQoS 2005, Passau, Germany, June 21-23, 2005. Proceedings 13*. Springer, 178–192.

[42] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 2012. Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication* (Helsinki, Finland) *(SIGCOMM '12)*. Association for Computing Machinery, New York, NY, USA, 443–454. https://doi.org/10.1145/2342356.2342440

[43] Goran Zuzic, Bernhard Haeupler, and Antti Roeyskoe. 2023. Sparse Semi-Oblivious Routing: Few Random Paths Suffice. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing* (Orlando, FL, USA) *(PODC '23)*. Association for Computing Machinery, New York, NY, USA, 222–232. https://doi.org/10.1145/3583668.3594585

Received 13-NOV-2023; accepted 2024-02-11