Breaking the VLB Barrier for Oblivious Reconfigurable Networks

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.

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 [FPR+10, WAK+10] and free-space optics [ZZZ+12, HQG+14, GMP+16] that enable reconfigurations within microseconds [PSF+13, LLF+14] or even nanoseconds [CWW+14, DWC+17]. Datacenter network architectures that leverage this capability are now being actively explored, including with recent prototype systems [GYG+17, SVB+19, MDG+20, BCB+20] and theoretical modeling and analysis [AWS+22, WAS+23, AAS23]. 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 [GYG+17], 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 [BKR03, HHR03, Räc02] and eventually logarithmic [Räc08] 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 [KCML05, ZSM05, BC07], 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 [AWS+22].

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 [GHMP08, BMB20]. Overprovisioning the network increases these costs proportionally [SVB+19], 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 only 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 Section 4 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

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.

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. Figure 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 $2g$ [AWS+22]. 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.

[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. Networks with degree constraint d can be simulated by networks with degree constraint 1, up to a slow-down by a factor of d [AWS+22]. Hence, the assumption that $d = 1$ is essentially without loss of generality, and we will continue adopting this assumption throughout the paper.

Goal	Average hop-count	Throughput	Reference
Minimize network hops	g	—	naïve counting
Uniform multicommodity flow	g	$\frac{1}{g}$	[AWS+22]
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)	$2g$	$\frac{1}{2g}$	[AWS+22]

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

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 2 for a visualization of these bounds. Additionally, like in [WAS+23] we condition against $\varepsilon = 1$. This is due to requiring a strictly positive slack factor between the throughput r and $\frac{1}{g+1}$.

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 $[N]$.

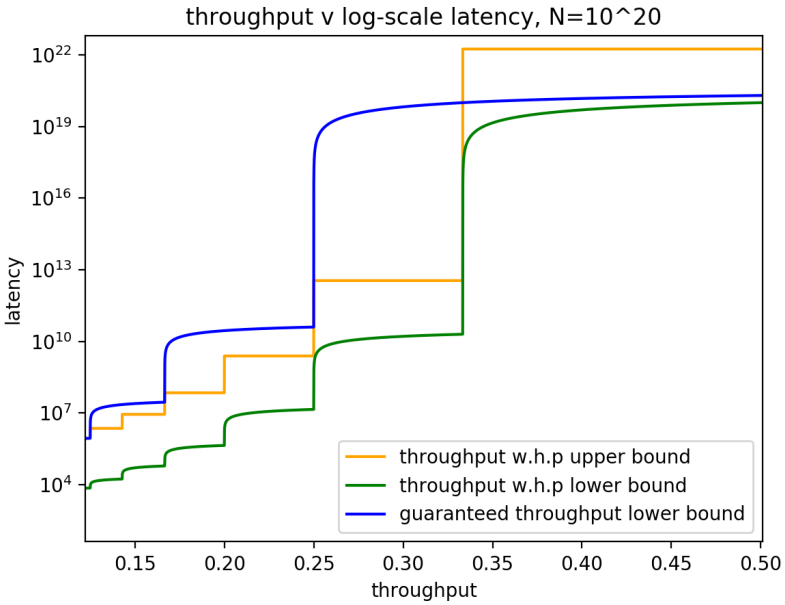

Figure 2: Throughput versus log-scale maximum latency tradeoff curves $\tilde{O}(L_{upp})$ and L_{low}, when compared against the guaranteed throughput lower bound of [AWS+22], on an ORN containing 10^{20} nodes.

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 [AWS+22] had 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 [DR96]. 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.

Our lower bound. Our lower bound proof is heavily inspired by the lower bound proof of [AWS+22]. 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, [AWS+22, WAS+23], 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. [Räc02] proved the existence of a polylog n-competitive algorithm for this problem, the competitive ratio later improved upon by [HHR03]. [BKR03, HHR03, ACF+03] then developed poly-time algorithms to achieve this result. Later, these algorithms were implemented and tested in wide-area networks [AC03]. [Räc08] 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 [FRT04]. This improved algorithm was again demonstrated in wide-area networks by [KYY+18].

Some works add additional constraints to this problem. For example, [GHR06] found a polylog n-competitive routing scheme oblivious to both traffic and the cost functions of edges, and [GHZ21] 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 [GHZ21]) 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. [KKT91, BH85]. 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 [VB81, Val82]. He later provided a lower bound in these contexts [Val83]. 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 [NJ94, NJ95]. 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.

[ALMN91] showed that in bit-serial routing, any random oblivious algorithm on a polylog degree network requires $\boldsymbol{O}(\log^2 n/\log\log n)$ bit-steps with high probability for almost all permutation traffic, assuming $\log n$-bit messages, extending the Borodin-Hopcroft bound for deterministic algorithms. [KYY+18] 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 [ZHR23]. 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 [ZHR23] constitutes the first provable asymptotic separation between semi-oblivious and oblivious routing in the literature, and the separation that we prove in Section 4.4 is the second such result.

A work that closely models the problem we ask [HKLR05], 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 [AWS+22] is first to name the ORN paradigm, it was used earlier in proposed network architectures and designs. Rotornet [GYG+17] and Sirius [BCB+20] both use

optical circuit switches to build a reconfigurable fabric, and Shoal [SVB+19] 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 [MDG+20] 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 [GZB+21] uses a modification of Rotornet as one component of an optical datacenter network, along with demand-aware reconfiguration and static graphs.

[AAS23] 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 these results, 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 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 *time-stationary demand* is a demand function in which every demand matrix $D(t)$ is the same. A *permutation demand* D_σ is a demand function in which every demand matrix is the permutation matrix defined by $\sigma : [N] \to [N]$. Note that permutation demands are also time-stationary.

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.

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 ε, 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.

3.1 Connection Schedule

The connection schedule of $\mathcal{R}^0$, like the Vandermonde Basis Scheme of [AWS+22], is based on round-robin phases (cf. Definition 13) defined by Vandermonde vectors. 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. 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 one period of the schedule 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 $\pi_k^\tau(a) = \tau^{-1}\Big(\pi_k^0\big(\tau(a)\big)\Big)$. 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 14. 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 $0 \le q \le g$, 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. (See Remark 1 below.) Instead, we prove the following novel tail bound for the distribution of bilinear sums on orbits of a permutation group action.

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

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

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

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

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 $\gamma > 0$,*
$$\Pr\left(B(\boldsymbol{u}, P\boldsymbol{v}) \;\ge\; e^\gamma \frac{\|\boldsymbol{u}\|_1 \, \|\boldsymbol{v}\|_1}{N}\right) \le M e^{-\frac{1}{2}\gamma^2 C}; \tag{6}$$

2. *for any $\gamma > 0$,*
$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \;\ge\; e^\gamma \frac{\|\boldsymbol{u}\|_1 \, \|\boldsymbol{v}\|_1}{N}\right) \le 15 M e^{-\frac{1}{100}\gamma^2 C}. \tag{7}$$

The proof of Theorem 3 is deferred to Section 3.4.

Proof of Theorem 2

Proof. 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 [AWS+22]. 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

[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.

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,

$$\begin{aligned}
X_i &= \sum_{j\in[N],\,j\neq i} \frac{\rho_q^-(i,e)\cdot\rho_{g-q}^+(e,j)}{\rho_{g+1}(i,j)}\cdot\left(\sum_{t\in I} D(t)_{ij}\right)\\
&\leq \sum_{j\in[N],\,j\neq i} \frac{\rho_q^-(i,e)\cdot\rho_{g-q}^+(e,j)\cdot rC(p-1)\cdot\overline{D}_{ij}}{(p-2)C^{g+1}}\\
&= \left(\frac{p-1}{p-2}\right) r \sum_{j\in[N],\,j\neq i} \overline{D}_{ij}\left(\frac{\rho_q^-(i,e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e,j)}{C^{g-q}}\right)\\
\sum_{i=1}^N X_i &\leq \left(\frac{p-1}{p-2}\right) r \sum_{i\neq j} \overline{D}_{ij}\left(\frac{\rho_q^-(i,e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e,j)}{C^{g-q}}\right) = \sum_{i\neq j}\overline{D}_{ij}u_iv_j \qquad (8)
\end{aligned}$$

where

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

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

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

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

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

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

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

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

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

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

$$\rho_q^-(i, \text{tail}(e)) \cdot \rho_q^+(\text{head}(e), j) \leq \max\{C^{q-1} \cdot C^{g-q},\, C^q \cdot C^{g-q-1}\} = C^{g-1}$$

$$\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty \leq \left(\frac{p-1}{p-2}\right) r \left(\frac{\rho_q^-(i, \text{tail}(e)) \cdot \rho_q^+(\text{head}(e), j)}{C^g}\right) \leq \left(\frac{p-1}{p-2}\right) \frac{r}{C}$$

$$\left(\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty}\right) \geq \frac{\frac{(p-1)^3}{p^2(p-2)} \cdot p^g \cdot r}{\frac{p-1}{p-2} \cdot \frac{r}{C}} = \left(\frac{p-1}{p}\right)^2 CN \geq \frac{1}{2} CN$$

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 $\gamma > 0$,

$$\Pr\left(B_\pi(\boldsymbol{u}, P\boldsymbol{v}) \geq e^\gamma r\right) \leq\; N^2 e^{-\frac{1}{4}\gamma^2 C}$$

$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^\gamma r\right) \leq 15 N^2 e^{-\frac{1}{200}\gamma^2 C}.$$

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

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

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

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

□

3.4 A Tail Bound for Bilinear Sums

In Section 3.3, our analysis of the distribution of the amount of flow traversing an edge e depends on certain tail bounds for the distribution of bilinear sums on orbits of a permutation group action. The relevant tail bound is stated as Theorem 3 above. This section is devoted to proving the theorem. The proof will make use of a Chernoff-type concentration bound for negatively associated random variables. We begin by recalling some definitions and facts about negative association; see [DR96, JDP83, Waj17] for an introduction to this topic.

Definition 15 ([JDP83, KLS81])**.** A set of random variables $X_1, ..., X_n$ are *negatively associated* if for any two functions $f, g : \mathbb{R}^n \to \mathbb{R}$ that are either both monotone increasing or both monotone decreasing, and dependent on[4] disjoint subsets of indices $S_f, S_g \subseteq [n]$, then

$$\mathbb{E}[f(\vec{X}) \cdot g(\vec{X})] \leq \mathbb{E}[f(\vec{X})] \cdot \mathbb{E}[g(\vec{X})]$$

Many examples of negatively associated random variables can be constructed using the following definition and lemma.

Definition 16. An n by m matrix A has *consistently ordered rows* if there exists some permutation $\pi : [m] \to [m]$ of the columns of A such that for all rows $i \in [n]$, $A[i, \pi(1)] \leq \cdots \leq A[i, \pi(m)]$.

Lemma 4. *Suppose A is an n by n matrix, and $X_1, ..., X_n$ are random variables sampled by the following process: sample a permutation $\pi : [n] \to [n]$ uniformly at random, and set $X_i = A[i, \pi(i)]$. If the entries of A are non-negative and A has consistently ordered rows, then $X_1, ..., X_n$ are negatively associated.*

[4]For the purposes of this definition, an n-variate function f is dependent on a set of indices $I \subseteq [n]$ if $f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$ holds whenever $x_i = y_i$ for all $i \in I$.

Proof. This will be proved by induction on n. Note that negative association amounts to showing that the covariance $Cov(f(\vec{X}), g(\vec{X})) \leq 0$. WLOG, since A has consistently ordered rows, we can assume that $A[i,1] \leq ..., A[i,n]$ for all $i \in [n]$.

Base case: $n = 2$. Then A is a 2 by 2 matrix, and since f,g are both either monotone increasing or monotone decreasing, then

$$Cov(f(\vec{X}), g(\vec{X})) = \frac{1}{4}\Big(f(A[1,1])g(A[2,1]) + f(A[1,2])g(A[2,2]) \\ - f(A[1,1])g(A[2,2]) - f(A[1,2])g(A[2,1])\Big) \leq 0$$

Now suppose the lemma is true for $n = k$, and for now suppose f, g are both monotone increasing. We will need two properties of covariance.

Property 1: (law of total covariance) Let X, Y, and Z be any random variables. Then $Cov(X,Y) = \mathbb{E}[Cov(X,Y)|Z] + Cov(\mathbb{E}[X|Z], \mathbb{E}[Y|Z])$.

Property 2: (Chebyshev's algebraic inequality) Given a random variable Z and monotone increasing h_1 and monotone decreasing h_2, then $Cov(h_1(Z), h_2(Z)) \leq 0$.

Now, consider the random variable $I = \pi^{-1}(1)$. This indicates which random variable X_i realizes its smallest value. Then by Property 1,

$$Cov(f(\vec{X}), g(\vec{X})) = \mathbb{E}\big[Cov(f(\vec{X}), g(\vec{X}))|I\big] + Cov\big(\mathbb{E}\big[f(\vec{X})|I\big], \mathbb{E}\big[g(\vec{X})|I\big]\big)$$

For any fixed I, the first term is random over 1 fewer variable, meaning this falls under the inductive hypothesis and is ≤ 0.

To show the second term is ≤ 0, we will show that as functions of I, one of $\mathbb{E}\big[f(\vec{X})|I\big]$ or $\mathbb{E}\big[g(\vec{X})|I\big]$ is monotone increasing, and the other is monotone decreasing.

Due to how the random variables X_i are chosen from A, they can be equivalently chosen from any matrix A' equivalent up to a re-ordering of rows. We will re-order the rows of A to enforce $h_1(I) = \mathbb{E}\big[f(\vec{X})|I\big]$ monotone increasing and $h_2(I) = \mathbb{E}\big[g(\vec{X})|I\big]$ monotone decreasing in I.

Let $\sigma_f : [|S_f|] \to S_f$ impose the ordering $h_1(\sigma_f(1)) \leq \ldots \leq h_1(\sigma_f(|S_f|))$. Additionally, let $\sigma_g : [|S_g|] \to S_g$ impose $h_2(\sigma_g(1)) \leq \ldots \leq h_2(\sigma_g(|S_g|))$.

Note that for $x \in S_f$, and $y \notin S_f$, then $\mathbb{E}\big[f(\vec{X})|I = x\big] \leq \mathbb{E}\big[f(\vec{X})|I = y\big]$, and the same holds true for g and S_g. We will re-order the rows of A in the following way: $\sigma_f(1), \ldots, \sigma_f(|S_f|)$, followed by all indices not within either S_f or S_g, followed by $\sigma_g(|S_g|), \ldots, \sigma_g(1)$. Then h_1 will be monotone increasing and h_2 will be monotone decreasing, thus showing $Cov(f(\vec{X}), g(\vec{X})) \leq 0$. An almost identical proof will show this true for f, g both monotone decreasing. □

Corollary 5. *If $\boldsymbol{u}$, $\boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-negative vectors, then the random variables $X_1, X_2, \ldots, X_N$ defined by sampling a uniformly random permutation $\pi : [N] \to [N]$ and setting $X_i = u_i v_{\pi(i)}$ are negatively associated.*

Proof. The matrix $A = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$ has non-negative entries and consistently ordered rows, so we may apply Lemma 4 to deduce the corollary. □

Corollary 6. *Let $\mathscr{X} = \{x_1, x_2, \ldots, x_m\}$ be any multiset of non-negative numbers, and for some $n \leq m$ let $X_1, X_2, \ldots, X_n$ denote random variables obtained by drawing n uniformly random samples* without replacement *from $\mathscr{X}$. (In other words, the conditional distribution of X_i given $X_1, \ldots, X_{i-1}$ is uniform over the multiset $\mathscr{X} \setminus \{X_1, \ldots, X_{i-1}\}$.) Then $X_1, \ldots, X_n$ are negatively associated.*

Proof. The special case $n = m$, in which the variables $X_1, \ldots, X_m$ constitute a random permutation of the elements of $\mathscr{X}$, can be obtained from Corollary 5 by setting $\boldsymbol{u} = (x_1, x_2, \ldots, x_m)^\mathsf{T}$ and $\boldsymbol{v} = (1, 1, \ldots, 1)^\mathsf{T}$. The general case in which $n \leq m$ can then be obtained by observing that the property of negative association is preserved under taking subsets of a set of random variables. □

We will be making use of the following Chernoff bound for negatively associated random variables.

Lemma 7. *Suppose $X_1, ..., X_N$ are negatively associated variables for which $X_i \in [0, 1]$ always, and $\mathbb{E}[\sum_i X_i] = \mu$. Then Chernoff's multiplicative tail bound holds. That is, for any $\gamma > 0$,*

$$\Pr\left[\sum_i X_i \geq e^\gamma \mu\right] \leq \left[\exp\left(e^\gamma - 1 - \gamma e^\gamma\right)\right]^\mu < e^{-\frac{1}{2}\gamma^2 \mu} \tag{10}$$

$$\Pr\left[\sum_i X_i \leq e^{-\gamma} \mu\right] \leq \left[\exp\left(e^{-\gamma} - 1 + \gamma e^{-\gamma}\right)\right]^\mu. \tag{11}$$

Furthermore, when $0 < \gamma < \frac{1}{2}$ the second inequality implies

$$\Pr\left[\sum_i X_i \leq e^{-\gamma} \mu\right] \leq e^{-\frac{1}{3}\gamma^2 \mu}. \tag{12}$$

The Chernoff bound is often expressed in terms of the tail probabilities $\Pr\left[\sum_i X_i \geq (1+\delta)\mu\right]$ and $\Pr\left[\sum_i X_i \leq (1-\delta)\mu\right]$, with the bound on the right side of the inequality then being written as a function of δ. For a proof, see [DR96, Waj17]. The version of the Chernoff bound stated above is obtained from the usual one by substituting $\gamma = \ln(1+\delta)$ in the first inequality and $\gamma = -\ln(1-\delta)$ in the second. The inequality $-e^\gamma + 1 + \gamma e^\gamma \geq \frac{1}{2}\gamma^2$ is derived by writing it in the equivalent form $\int_0^\gamma te^t\, dt \geq \int_0^\gamma t\, dt$ and comparing integrands. The inequality $-e^{-\gamma} + 1 - \gamma e^{-\gamma} \geq \frac{1}{3}\gamma^2$ is justified by using Taylor's Theorem to deduce that the left side is bounded below by $\frac{1}{2}\gamma^2 - \frac{1}{3}\gamma^3$ for $0 < \gamma < 1$ and then noting that $\frac{1}{2}\gamma^2 \geq \frac{1}{3}\gamma^2 + \frac{1}{3}\gamma^3$ when $0 < \gamma < \frac{1}{2}$.

As a first application of Lemma 7 we can prove the first tail bound asserted in Theorem 3.

Lemma 8. *Suppose $\boldsymbol{u}, \boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying*

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

Suppose D is a doubly stochastic matrix defining a bilinear form $B(\cdot, \cdot)$ via

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

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 for any $\gamma > 0$,

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

Proof. The Birkhoff-von Neumann Theorem says that D can be expressed as a convex combination of permutation matrices, and Carathéodory's Theorem says that there exists such an expression in which the number of constituent permutation matrices is at most $(N-1)^2 + 1$, which is bounded

above by N^2. Hence, D can be expressed as a convex combination of at most M permutation matrices, where M is defined as in the lemma statement. The bilinear form B is thus a convex combination of at most M bilinear forms B_σ, where B_σ is defined for a permutation σ by

$$B_\sigma(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i:\sigma(i)\neq i} u_i v_{\sigma(i)}.$$

We will prove the special case of the lemma when D is a permutation matrix and $B = B_\sigma$ for some σ; the general case will then follow by the union bound.

If τ is the random permutation such that $P_{i,\tau(i)} = 1$ for all i, then for any permutation σ the composition $\pi = \tau \circ \sigma$ is uniformly distributed over all permutations of $[N]$. Consequently, by Corollary 5, the random variables $X_i = \frac{u_i v_{\pi(i)}}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty}$ are negatively associated. By construction, they take values between 0 and 1. Furthermore, the expected value of $\sum_{i=1}^N X_i$ can be computed by linearity of expectation, using the fact that the event $\pi(i) = j$ has probability $\frac{1}{N}$ for all j.

$$\mu = \mathbb{E}\left[\sum_{i=1}^N X_i\right] = \frac{1}{N}\sum_{i=1}^N \sum_{j=1}^N \frac{u_i v_j}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty} = \frac{1}{N}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty} \geq C.$$

Applying Lemma 7, the probability that $\sum_{i=1}^N X_i$ exceeds $\frac{e^\gamma}{N}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty \boldsymbol{u}\|_\infty}$ is less than $e^{-(1/2)\gamma^2 C}$. Inequality (6) follows because $B_\sigma(\boldsymbol{u}, \boldsymbol{v}) \leq \|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty \sum_{i=1}^N X_i$. □

Remark 1. After seeing the proof of the tail bound (6), it is tempting to try proving an analogous tail bound for $B(P\boldsymbol{u}, P\boldsymbol{v})$ using the random variables $X_1, \ldots, X_N$ defined by

$$X_i = \frac{u_{\tau(i)} v_{\tau(\sigma(i))}}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty}.$$

The trouble is that these random variables may fail to be negatively associated. As a simple example, suppose $\boldsymbol{u} = \boldsymbol{v} = (1, 1, 0, 0)^\mathsf{T}$ and let $\sigma = (1\ 2)(3\ 4)$ be the permutation of $\{1, 2, 3, 4\}$ that transposes the first and last pairs of elements. Then $X_1 = u_{\tau(1)} v_{\tau(2)}$ and $X_2 = u_{\tau(2)} v_{\tau(1)}$. When $\tau(\{1, 2\}) = \{1, 2\}$ we have $X_1 = X_2 = 1$, and otherwise $X_1 = X_2 = 0$. Hence, $\mathbb{E}[X_1 X_2] = \frac{1}{6} > \mathbb{E}[X_1]\mathbb{E}[X_2]$, violating negative association.

Despite the counterexample in Remark 1, we will still be able to prove a tail bound for $B(P\boldsymbol{u}, P\boldsymbol{v})$ using negative association and the Chernoff bound, however we will need to pursue a more indirect strategy. We begin with the following tail bound for random submatrices of a non-negative rank-one matrix.

Lemma 9. *Suppose $\boldsymbol{u}, \boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying (4). For any $K \leq N/2$ let (Q, R) denote a uniformly random sample from the set of ordered pairs of K-element subsets of $[N]$ that are disjoint from one another. Then for $0 < \gamma < 1$,*

$$\Pr\left(\sum_{i\in Q}\sum_{j\in R} u_i v_j \;\geq\; e^\gamma \frac{K^2}{N^2}\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1\right) \;\leq\; 2e^{-\frac{1}{8}\gamma^2 CK/N} \tag{13}$$

$$\Pr\left(\sum_{i\in Q}\sum_{j\in R} u_i v_j \;\leq\; e^{-\gamma} \frac{K^2}{N^2}\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1\right) \;\leq\; 2e^{-\frac{1}{12}\gamma^2 CK/N}. \tag{14}$$

Proof. If $\sum_{i\in Q}\sum_{j\in R} u_i v_j \geq e^{\gamma}\frac{K^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1$ then at least one of the inequalities

$$\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty} \geq e^{\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \tag{15}$$

$$\sum_{j\in R}\frac{v_i}{\|\boldsymbol{v}\|_\infty} \geq e^{\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \tag{16}$$

is satisfied. Similarly, if $\sum_{i\in Q}\sum_{j\in R} u_i v_j \leq e^{-\gamma}\frac{K^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1$ then at least one of the inequalities

$$\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty} \leq e^{-\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \tag{17}$$

$$\sum_{j\in R}\frac{v_i}{\|\boldsymbol{v}\|_\infty} \leq e^{-\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \tag{18}$$

is satisfied. To bound the probabilities of these events, let $X_1, X_2, \ldots, X_K$ be random variables obtained by drawing K uniformly random samples without replacement from the multiset $\{\frac{u_i}{\|\boldsymbol{u}\|_\infty} \mid 1 \leq i \leq n\}$ and observe that $X_1 + \cdots + X_K$ and $\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty}$ are identically distributed. By Corollary 6 the random variables $X_1, \ldots, X_K$ are negatively associated, by construction they are $[0,1]$-valued, and by linearity of expectation their sum has expected value $\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}$. The assumption that $\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \geq CN$, combined with the inequality $\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \leq N$, implies $\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \geq CK/N$. The Chernoff bound now implies that the probability of inequality (15) being satisfied is at most $e^{-\frac{1}{8}\gamma^2 CK/N}$, and the probability of inequality (17) being satisfied is at most $e^{-\frac{1}{12}\gamma^2 CK/N}$. A similar argument using K random variables drawn without replacement from the multiset $\{\frac{v_i}{\|\boldsymbol{v}\|_\infty} \mid 1 \leq i \leq n\}$ establishes that the probabilities of inequalities (16) and (18) being satisfied are bounded above by $e^{-\frac{1}{8}\gamma^2 CK/N}$ and $e^{-\frac{1}{12}\gamma^2 CK/N}$, respectively. The lemma now follows by applying the union bound. □

We are now ready to restate and prove Theorem 3.

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 $\gamma > 0$,*
$$\Pr\left(B(\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1\,\|\boldsymbol{v}\|_1}{N}\right) \leq Me^{-\frac{1}{2}\gamma^2 C}; \tag{6}$$
2. *for any $\gamma > 0$,*
$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1\,\|\boldsymbol{v}\|_1}{N}\right) \leq 15Me^{-\frac{1}{100}\gamma^2 C}. \tag{7}$$

Proof. The first tail bound, inequality (6), was already proven in Lemma 8, so turn to proving (7). As in the proof of (6), we will be using the Birkhoff-von Neumann Theorem, Carathéodory's Theorem, and the union bound to reduce to the case where the doubly stochastic matrix D is a permutation matrix. Accordingly, for the remainder of the proof we will be focused on a fixed permutation σ and its associated bilinear form

$$B_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i:\sigma(i)\neq i} x_i y_{\sigma(i)},$$

and our goal will be to prove the tail bound (7) when $B = B_\sigma$ and $M = 1$.

To start, we note that it is without loss of generality to assume that σ has at most one fixed point. This is because if F is the fixed-point set of σ and $|F| > 1$, then we can compose σ with a permutation whose fixed-point-set is the complement of F, to obtain a fixed-point-free permutation $\tilde{\sigma}$ that agrees with σ on the complement of F. For every pair of non-negative vectors $\boldsymbol{x}, \boldsymbol{y}$ we have $B_{\tilde{\sigma}}(\boldsymbol{x}, \boldsymbol{y}) \geq B_\sigma(\boldsymbol{x}, \boldsymbol{y})$, so an upper bound on the probability of $B_{\tilde{\sigma}}(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^\gamma \frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}$ will suffice to prove an upper bound on the probability of the same event for B_σ. Henceforth we will ignore the distinction between σ and $\tilde{\sigma}$ and we'll simply assume that σ has at most one fixed point. Let N^* denote the complement of F in $[N]$, i.e. $N^* = \{i \mid \sigma(i) \neq i\}$.

Define the *cycle diagram* of σ to be the directed graph with vertex set N^* and edge set $\{(i, \sigma(i)) \mid i \in N^*\}$, which is a disjoint union of directed cycles. The next step of the proof is to define a *balanced 3-coloring* $\chi : N^* \to \{0, 1, 2\}$ of the cycle diagram of σ, by which we mean a proper 3-coloring such that each color is used at least $\lfloor |N^*|/3 \rfloor$ and at most $\lceil |N^*|/3 \rceil$ times. We will then break down the bilinear form B_σ as a sum $B_\sigma^{(0)} + B_\sigma^{(1)} + B_\sigma^{(1)}$ where for $q \in \{0, 1, 2\}$,

$$B_\sigma^{(q)}(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i:\chi(i)=q} u_i v_{\sigma(i)},$$

and we will prove exponential tail bounds for each of the quantities $B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v})$. The purpose of the 3-coloring is to allow us to condition on an event that breaks up dependencies such as the one identified in Remark 1, enabling the use of negative association and Chernoff bounds.

One can find a balanced coloring of the cycle diagram of σ by a greedy strategy, combining the following two simple observations.

1. *Every directed 2-cycle has a balanced 3-coloring.* If the cycle has length k, then color the i^{th} vertex of the cycle with $\chi(i) = i \pmod 3$ unless $k \equiv 1 \pmod 3$, in which case the first $k - 1$ vertices of the cycle are colored using $\chi(i) = i \pmod 3$ and the last vertex is colored with the unique color that differs from both of its neighbors' colors.

2. *The disjoint union of two graphs with balanced 3-colorings also has a balanced 3-coloring.* If a balanced 3-coloring of a graph with n vertices, let us say that a color is *overused* if it is used more than $\lfloor n/3 \rfloor$ times. If graph G is the disjoint union of G_0 and G_1, each of which has a balanced 3-coloring, let k_0 and k_1 denote the number of overused colors in G_0 and G_1, respectively. If $k_0 + k_1 \leq 3$ then we can recolor G_1 if necessary so that its set of overused colors is disjoint from the set of overused colors in G_0. The union of the two colorings is then a balanced 3-coloring of G. If $k_0 + k_1 > 3$ then it must be the case that $k_0 = k_1 = 2$, in which case we can recolor G_1 if necessary so that each $q \in \{0, 1, 2\}$ is overused in at least one of G_0, G_1, and exactly one color is overused in both. The union of the two colorings is then a balanced coloring of G.

Having defined the coloring χ we now focus on one specific color $q \in \{0,1,2\}$ and aim to prove a tail bound for $B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v})$ when τ is a uniformly random permutation and P is the permutation matrix with $P_{i,\tau(i)} = 1$ for all i. To do so we will define $I = \chi^{-1}(\{q\})$ to be the set of indices $i \in N^*$ whose color is q, and we will condition on the random variable $Z = \tau|_{N^* \setminus I}$, the restriction of τ to indices whose color differs from q. Some useful observations are the following.

[O1] The set $Q = \tau(I)$ is uniquely determined by Z: it is equal to the complement of $\tau(N^* \setminus I)$ in N^*.

[O2] The set $R = \tau(\sigma(I))$ is also uniquely determined by Z. In fact, because $\chi(\sigma(i)) \neq \chi(i)$ for all i, the set $\sigma(I)$ must be disjoint from I, so the value of $\tau(i)$ for each $i \in \sigma(I)$ is determined by Z.

[O3] Let $K_q = |I|$. Since I and $\sigma(I)$ are disjoint K_q-element subsets of N^* and τ is a uniformly random permutation of N^*, the joint distribution of the pair of sets $(Q, R) = (\tau(I), \tau(\sigma(I))$ is the uniform distribution on ordered pairs of disjoint K_q-element subsets of N^*.

[O4] Conditional on Z, the restriction of τ to I is a uniformly random bijection between I and Q.

Define a random variable Y by

$$Y = \sum_{j \in Q} \sum_{k \in R} u_j v_k$$

and observe that the value of Y is determined by Z, since Z determines the sets Q and R. By Observation **[O4]** and linearity of expectation we have

$$\mathbb{E}[B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v}) \mid Z] = \sum_{i \in I} \mathbb{E}[u_{\tau(i)} v_{\tau(\sigma(i))} \mid Z] = \frac{1}{K_q} \sum_{j \in Q} \sum_{k \in R} u_j v_k = \frac{Y}{K_q}. \tag{19}$$

Our goal now turns to bounding the probabilities of the following "bad events."

$$\mathcal{E}_1^q = \left\{ Y \leq e^{-\frac{5}{7}\gamma} \frac{K_q^2}{N^2} \|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1 \right\}$$
$$\mathcal{E}_2^q = \left\{ Y \geq e^{\frac{4}{7}\gamma} \frac{K_q^2}{N^2} \|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1 \right\}$$
$$\mathcal{E}_3^q = \left\{ B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\frac{3}{7}\gamma} \frac{Y}{K_q} \right\}.$$

First, using Lemma 9 and the inequality $K/N \geq \frac{1}{N}\lfloor (N-1)/3 \rfloor \geq 1/4$, we have

$$\Pr(\mathcal{E}_1^q) \leq 2e^{-\frac{1}{12}\left(\frac{5\gamma}{7}\right)^2 \frac{C}{4}} < 2e^{-\frac{1}{100}\gamma^2 C}, \qquad \Pr(\mathcal{E}_2^q) \leq 2e^{-\frac{1}{8}\left(\frac{4\gamma}{7}\right)^2 \frac{C}{4}} < 2e^{-\frac{1}{100}\gamma^2 C}. \tag{20}$$

Next we turn to bounding the conditional probability $\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q \mid Z = z)$, for each value z in the support of Z. Recall that the value of Y is determined by Z, and the event $\mathcal{E}_1^q$ is determined by the value of Y. Hence, the values z in the support of Z may be partitioned into two sets: $\mathcal{Z}_0$ is the set of z such that $\mathcal{E}_1^q$ does not occur when $Z = z$, and $\mathcal{Z}_1$ is the set of z such that $\mathcal{E}_1^q$ occurs when $Z = z$. Obviously, for $z \in \mathcal{Z}_1$, $\Pr(\mathcal{E}_1^q \mid Z = z) = 1$ so $\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q \mid Z = z) = 0$.

Assume henceforth that $z \in \mathcal{Z}_0$. Then $Y > e^{-\frac{5}{7}\gamma}\frac{K_q^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1$. Now, let $\boldsymbol{u}_Q$ denote the subvector of $\boldsymbol{u}$ indexed by the elements of Q, and let $\boldsymbol{v}_R$ denote the subvector of $\boldsymbol{v}$ indexed by the elements of R. We will apply Lemma 8 to this pair of vectors. Note that $\|\boldsymbol{u}_Q\|_1\|\boldsymbol{v}_R\|_1 = Y$. Hence,

$$\frac{\|\boldsymbol{u}_Q\|_1\|\boldsymbol{v}_R\|_1}{\|\boldsymbol{u}_Q\|_\infty\|\boldsymbol{v}_R\|_\infty} \geq \frac{Y}{\|\boldsymbol{u}\|_\infty\|\boldsymbol{v}\|_\infty}$$
$$> e^{-\frac{5}{7}\gamma}\frac{K_q^2}{N^2}\frac{\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty\|\boldsymbol{v}\|_\infty} \quad \geq e^{-\frac{5}{7}\gamma}\frac{K_q^2}{N^2}\cdot CN > \frac{e^{-\frac{5}{7}}K_q}{N}\cdot CK_q > \frac{e^{-\frac{5}{7}}}{4}CK_q > \frac{C}{9}K_q.$$

By Observation **[O4]**, the random variable $B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v})$ can be calculated by sampling a uniformly random bijection π between Q and R and computing the sum $\sum_{i\in Q} u_i v_{\pi(i)}$. Hence, by Lemma 8,

$$\Pr(\mathcal{E}_3^q \mid Z = z \in \mathcal{Z}_0) \leq e^{-\frac{1}{2}(\frac{3}{7}\gamma)^2\frac{C}{9}} < e^{-\frac{1}{100}\gamma^2 C}. \tag{21}$$

Combining the cases $z \in \mathcal{Z}_0$ and $z \in \mathcal{Z}_1$, we have proven that $\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q \mid Z) < e^{-\frac{1}{100}\gamma^2 C}$ pointwise. Hence,

$$\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q) = \mathbb{E}_Z\left[\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q \mid Z)\right] < e^{-\frac{1}{100}\gamma^2 C}.$$

Now, by the union bound, we find that

$$\Pr(\mathcal{E}_2^q \cup \mathcal{E}_3^q) \leq \Pr(\mathcal{E}_1^q \cup \mathcal{E}_2^q \cup \mathcal{E}_3^q) \leq \Pr(\mathcal{E}_1^q) + \Pr(\mathcal{E}_2^q) + \Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q) \leq 5e^{-\frac{1}{100}\gamma^2 C}.$$

On the complement of $\mathcal{E}_2^q \cup \mathcal{E}_3^q$, we have the inequalities

$$B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v}) < e^{\frac{3}{5}\gamma}\frac{Y}{K_q} < e^{\frac{3}{5}\gamma}\cdot e^{\frac{2}{5}\gamma}\cdot\frac{1}{K_q}\cdot\frac{K_q^2}{N^2}\cdot\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1 = e^\gamma\frac{K_q}{N}\cdot\frac{\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1}{N}. \tag{22}$$

With probability at least $1 - 15e^{-\frac{1}{100}\gamma^2 C}$, the event $\mathcal{E}_2^q \cup \mathcal{E}_3^q$ does not occur for any $q \in \{0,1,2\}$. In that case,

$$B_\sigma(P\boldsymbol{u}, P\boldsymbol{v}) = B_\sigma^{(0)}(P\boldsymbol{u}, P\boldsymbol{v}) + B_\sigma^{(1)}(P\boldsymbol{u}, P\boldsymbol{v}) + B_\sigma^{(2)}(P\boldsymbol{u}, P\boldsymbol{v}) < e^\gamma\frac{K_0 + K_1 + K_2}{N}\cdot\frac{\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1}{N} \leq e^\gamma\frac{\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1}{N}. \tag{23}$$

Hence, the negation of this inequality occurs with probability at most $15e^{-\frac{1}{100}\gamma^2 C}$, as claimed. □

4 Upper Bound: Semi-Oblivious Design

In this section we prove Theorem 1.3, restated below.

Theorem 1.3. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor\frac{1}{r} - 1\rfloor$, and let*

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

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

Similar to Section 3, we will begin by constructing an SORN design $\mathcal{S}^0$ which is parameterized by N, g, and C, where C is a parameter which we set during our analysis to achieve the appropriate tradeoffs between throughput and latency. We will then analyze $\mathscr{S}_N(g, C)$, a distribution over all SORN designs $\mathcal{S}^\tau$ which are equivalent to $\mathcal{S}^0$ up to re-labeling of nodes, and show that it satisfies the conclusion of Theorem 1.3. Before we define $\mathcal{S}^0$, we first provide some intuition behind the design.

Definition 17. 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 $\mathbb{F}_p^g$.

The ORN design described in Section 3 was defined using phases of Vandermonde vectors. This was only done to achieve the property that any set of g vectors, each chosen from a different phase block, formed a basis over $\mathbb{F}_p^g$. No other special property of Vandermonde vectors was required. Thus, using any (C, g)-constellation gives the same throughput-latency tradeoffs found in Theorem 1.

In order to guarantee throughput r rather than achieve it with high probability, we need to provide alternate routing paths in the low probability case that the network becomes congested. We will do this by rotating through a series of different (C, g)-constellations, so that in an entire period of the schedule, each node is directly connected to most other nodes an equal number of times. Our alternate paths will then use a simple 2-hop Valiant load balancing (VLB) routing strategy.

Lemma 10. *Suppose $A \in \mathbb{F}_p^{g\times g}$ is an invertible matrix, and $\mathcal{V} = (v_1, v_2, \ldots, v_{C(g+1)})$ is a (C, g)-constellation in $\mathbb{F}_p^g$. Then the sequence $A\mathcal{V} = (Av_1, Av_2, \ldots, Av_{C(g+1)})$ is also a (C, g)-constellation in $\mathbb{F}_p^g$.*

Proof. Suppose not, that there exists some set of vectors $w_{i_1}, \ldots w_{i_g}$ each from different blocks of $A\mathcal{V}$ which are linearly dependent. Then WLOG there exists constants $\alpha_1, \ldots, \alpha_{g-1}$ such that $\alpha_1 w_{i_1} + \ldots + \alpha_{g-1} w_{i_{g-1}} = w_{i_g}$. Then $\alpha_1 A v_{i_1} + \ldots + \alpha_{g-1} A v_{i_{g-1}} = A v_{i_g}$ for vectors $v_{i_1}, \ldots, v_{i_g}$ each from different blocks of $\mathcal{V}$. This is a contradiction due to distributivity of matrix and vector multiplication, and because A is invertible. □

4.1 Connection Schedule

We now move to defining the connection schedule of $\mathcal{S}^0$. Consider the set of all diagonal invertible matrices $\mathcal{M}$, and let two matrices M_1, M_2 be related by $\sim$ if they are scalar multiples of one another. That is, if there is some scalar $a \in \mathbb{F}_p$ such that $M_1 = aM_2$. Let $\mathcal{A} \subset \mathcal{M}$ contain one representative from each of the equivalence classes of $\sim$. (Note that therefore, $|\mathcal{A}| = (p-1)^{g-1}$.) Also let $\mathcal{V}$ be any sequence of $C(g+1)$ distinct Vandermonde vectors not including the vector $(1, 0, \ldots, 0)$. Order $\mathcal{V}$ arbitrarily, so that $\mathcal{V} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{C(g+1)-1}\}$.

Then by Lemma 10, $A\mathcal{V}$ is a (C, g)-constellation for any matrix $A \in \mathcal{A}$. Order the set of matrices $\mathcal{A}$ arbitrarily, so that $\mathcal{A} = \{A_0, A_1, \ldots, A_{(p-1)^{g-1}-1}\}$. We rotate through the (C, g)-constellations formed by matrices in $\mathcal{A}$ to achieve our connection schedule.

More formally, we set the period length of the schedule to be $T = (p-1)^{g-1}C(g+1)(p-1) = (p-1)^g C(g+1) < C(g+1)N$, and we identify each congruence class $k \pmod T$ with a constellation number f, a phase number x and a scale factor s, for which $0 \le f \le (p-1)^{g-1} - 1$, $0 \le x < C(g+1)$, and $1 \le s < p$, such that $k = C(g+1)(p-1)f + (p-1)x + s - 1$. It is useful to think of timesteps as 3-tuples, $k = (f, x, s)$, so we will sometimes abuse notation and refer to timestep (f, x, s) in the sequel, when we mean $k = C(g+1)(p-1)f + (p-1)x + s - 1$. The connection schedule of $\mathcal{R}^0$, during timesteps $t \equiv k \pmod T$, uses permutation $\pi_k^0(a) = a + sA_f\boldsymbol{v}_x$, where f, x and s are the constellation number, phase number, and scale associated to k.

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

4.2 Routing Protocol

The routing protocol $\{S^0_\sigma : \sigma \text{ permut on } [N]\}$ will, for each σ, use one of two types of routing paths. The first type is the $(g+1)$-hop paths that we wish to route on. For most σ, routing on these paths will not overload any edges in the network. Thus, for those σ, S^0_σ will include only those such paths.

However, with low probability over σ, routing on these paths will cause too much congestion on some edge in the network to be used. In this case, we will designate an alternate set of paths for S^0_σ to use. The alternate set of paths will take only 2 hops in the network, and will suffer significantly higher maximum latency. However, we will show that since this is a low probability event over choice of σ, this will not meaningfully increase our average latency.

To route from node a to node b starting at timestep t, first delay until a new (C, g)-constellation $A\mathcal{V}$ begins.

$(g+1)$-hop paths. In this case, we use the same distribution over routing paths as in Section 3.2, when considering the set of $C(g+1)$ phases all belonging to the (C, g)-constellation beginning after time t. Due to the added delay, paths of this type have maximum latency $2C(g+1)N^{1/h}$, instead of the maximum latency cited in Section 3.2.

2-hop paths. To describe the distribution over 2-hop paths, first consider the following. Given a fixed Vandermonde vector $\boldsymbol{v} \in \mathcal{V}$, consider the set of edges formed by $a \to a + sA\boldsymbol{v} = b$ for all scalar factors s and matrices $A \in \mathcal{A}$. Note that an edge between any node pair a, b for which the vector $b - a$ has only non-zero coordinates appears exactly once in this set. This is because $A \in \mathcal{A}$ contains all invertible diagonal matrices which are not scalar multiples of each other. Additionally, an edge between a, b never appears if the vector $b - a$ has any coordinates equal to zero. (Recall the vector $(1, 0, \ldots, 0) \notin \mathcal{V}$.) Then across the entire period, an edge between any node pair a, b for which the vector $b - a$ has only non-zero coordinates appears exactly $C(g+1)$ times, once for each $\boldsymbol{v} \in \mathcal{V}$.

Consider the following random process for choosing a 2-hop path from a to b. Uniformly at random, choose a node b' for which both $b' - a$ and $b - b'$ have only non-zero coordinates. Also uniformly at random, choose Vandermonde vectors $v_a, v_b \in \mathcal{V}$. Compute the unique invertible diagonal matrices $A_a, A_b \in \mathcal{A}$ and scalar factors $s_a, s_b \in \{1, \ldots, p-1\}$ for which $b' - a = s_a A_a v_a$ and $b - b' = s_b A_b v_b$. Over the next full period of the schedule, or $(p-1)^{g-1}C(g+1)(p-1)$ timesteps, take the direct hop from a to b' which appears during the (C, g)-constellation $A_a\mathcal{V}$. Wait for the period to finish. Then during the next period, take the hop from b' to b which appears during the (C, g)-constellation $A_b\mathcal{V}$.

Note that paths of this type always take both hops during consecutive distinct periods, or iterations, of the schedule. Thus, paths of this type will have maximum latency $2(p-1)^g C(g+1) + C(g+1)(p-1) \leq C(g+1)N^{1/h} + 2C(g+1)N \leq \tilde{O}(N)$.

If routing rD_σ on $(g+1)$-hop paths does not overload edges in the network, then S_σ routes all demand between $a, \sigma(a)$ pairs on $(g+1)$-hop paths. Otherwise, if routing rD_σ on $(g+1)$-hop paths would overload some edge in the network, then S_σ routes all demand between $a, \sigma(a)$ pairs on 2-hop paths.

As written here, S^0_σ must make one choice for all timesteps t: to either route on $(g+1)$-hop paths or 2-hop paths. In Appendix A, we discuss how to analyze a design which allows S^0_σ to route flow on a combination of $(g+1)$-hop and 2-hop paths, depending on starting timestep t.

To route over $\mathcal{S}^\tau$ for general τ, note that the edges of $\mathcal{S}^\tau$ are in a bijection with $\mathcal{S}^0$. Thus, any path from node a to node b in $\mathcal{S}^\tau$ has a unique equivalent path from $\tau(a)$ to $\tau(b)$ in $\mathcal{S}^0$. To define the routing protocol $\{S^\tau_\sigma : \sigma \text{ permut on } [N]\}$ in $\mathcal{S}^\tau$, simply apply this bijection to the routing paths from $\tau(a)$ to $\tau(\sigma(a))$ in $\{S^0_\sigma : \sigma \text{ permut on } [N]\}$.

4.3 Throughput-Latency Tradeoff

Theorem 11. *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 $\max\left\{C(g+1), 2 + \frac{2}{1-\varepsilon}, \frac{g+3}{\varepsilon} - 2, \frac{2-\delta}{1-\delta}\right\}$ for $\delta = \frac{(g+1)^{1/g}}{(g+2-\varepsilon)^{1/g}}$, let $\gamma = \ln\left(\frac{g+2-\varepsilon}{g+1}\right)$ and $C = \frac{\log\log N}{\gamma^2}\ln(N)$, and let*

$$L_{upp} = gN^{1/g}$$

Then:

1. *the fixed SORN design $\mathcal{S}^0$ guarantees throughput r (with respect to stationary demands), and achieves maximum latency $\tilde{\boldsymbol{O}}(L_{upp})$ with high probability under the uniform distribution.*
2. *the family of distributions $\mathscr{S}_N(g, C)$ guarantees throughput r, and achieves maximum latency $\tilde{\boldsymbol{O}}(L_{upp})$ with high probability.*

Note that if $\varepsilon = 1$, then $\frac{1}{r} \in \mathbb{Z}$, and there do not exist primes p for which $p \geq 2 + \frac{2}{1-\varepsilon}$. Thus, we condition against $\varepsilon = 1$.

Both parts of this theorem will be proven by focusing on the probability that S^0_σ must deviate from sending all flow on $(g+1)$-hop paths to sending all flow on 2-hop paths. This is directly correlated with when congestion occurs on physical edges in the design $\mathcal{S}^0$, if we were to always send flow on $(g+1)$-hop paths. We note the similarities between $\mathcal{S}^0$ and $\mathcal{R}^0$ from Section 3, and apply the same exponential tail bounds of bilinear sums to get our result.

Proof. First, let us confirm that the 2-hop "failover scheme" of $\mathcal{S}^\tau$ guarantees throughput r. Fix some permutation demand D_σ and an edge e, and consider for each demand pair $i, \sigma(i)$ how much much flow is crossing edge e due to $i, \sigma(i)$ traveling on 2-hop paths. If 1st hop flow crosses edge e from i to $\sigma(i)$, then it must be the case that $tail(e) = i$ and both $head(e) - i$ and $\sigma(i) - head(e)$ have only non-zero coordinates. Similarly, if 2nd hop flow crosses edge e from i to $\sigma(i)$, then $head(e) = \sigma(i)$ and both $tail(e) - i$ and $\sigma(i) - tail(e)$ have only non-zero coordinates.

Each demand pair $i, \sigma(i)$ contributes $rC(g+1)(p-1)^g$ total flow per period. For any node pair $i, \sigma(i)$, there are at least $(p-2)^g$ different nodes b for which $b - i$ and $\sigma(1) - b$ both have only non-zero coordinates. And for each of these nodes b, there are exactly C different phases which connect i to b, and exactly C different phases which connect b to $\sigma(i)$. Thus, the amount of first-hop flow traversing edge e is no more than $\frac{rC(g+1)(p-1)^g}{C(p-2)^g}$. This is no more than 1 when $p \geq \frac{2-\delta}{1-\delta}$ for $\delta = \frac{(g+1)^{1/g}}{(g+2-\varepsilon)^{1/g}}$, which we condition on in the statement of the theorem.

Thus, we focus on showing that $\mathcal{S}^0$ sends flow on $(g+1)$-hop paths with high probability over the uniform distribution.

Like before, we may assume without loss of generality that the demand matrix $D(t)$ is doubly stochastic for all t.

We first consider the failure probability of edges within each (C, g)-constellation individually. 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 of the flow path.

Note that, unlike in the proof of Theorem 2, edges e that appear in the $(q+1)$th phase block of a (C, g)-constellation, for $0 \leq q \leq g$, will *only* have $(q+1)$-th hop flow traversing e, due to delaying flow before routing by whole (C, g)-constellations instead of single phase blocks. Then the total amount of $(q+1)$-th hop flow traversing edge e equals the total amount of any-hop flow traversing edge e.

First we examine $q = 0$. First-hop flow traversing edge e originates at source node tail(e) during the constellation preceding the one to which e belongs. There are $C(g+1)(p-1)$ 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 $rC(g+1)(p-1)$ 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, for an edge e occurring in the first phase block of a (C, g)-constellation, the amount of first-hop flow on e is bounded above by $\frac{rC(g+1)(p-1)\cdot C^g}{(p-2)C^{g+1}} = \left(\frac{p-1}{p-2}\right)(g+1)r$. (Note that this is not a probabilistic statement; the upper bound on first-hop flow holds with probability 1.) A symmetric argument shows that for an edge e occurring in the last phase block of a (C, g)-constellation, the amount of last-hop flow on e is bounded above by $\left(\frac{p-1}{p-2}\right)(g+1)r$ as well.

Now suppose $1 \leq q \leq g-1$, and let Y_i be the random variable realizing the amount of $(q+1)$-th hop flow traversing edge e due to source node i, normalized by $\frac{1}{g+1}$. Clearly, the total amount of $(q+1)$-th hop flow traversing e will be $(g+1)\sum_i Y_i$. The variables Y_i act exactly as the random variables X_i in Section 3.3, in the proof of Theorem 2. Therefore, the same tail bound conclusions about their sum are applicable.

Therefore, over the uniform distribution for the fixed design $\mathcal{S}^0$, and for the family of distributions $\mathscr{S}_N(g, C)$, we have

$$
\begin{aligned}
&\Pr[e \text{ has } \geq (g+1)e^{\gamma}r \text{ flow when routing } (g+1)\text{-hop paths}] \leq 15N^2 e^{-\frac{1}{200}\gamma^2 C} \\
\implies &\Pr[\text{any edge } e \text{ has } \geq (g+1)e^{\gamma}r \text{ flow when routing } (g+1)\text{-hop paths}] \\
&\leq (p-1)^{g-1}C(g+1)(p-1)N \cdot 15N^2 \left(e^{-\frac{1}{200}\gamma^2}\right)^C \\
&\leq 15N^4(g+1)\frac{\log\log N}{\gamma^2}\ln(N)e^{-\frac{1}{200}\log\log N \ln(N)} \\
&\leq \left(15N^4(g+1)\frac{\log\log N \ln(N)}{\gamma^2}\right) N^{-\frac{1}{200}\log\log N} \\
&\leq O\left(\frac{1}{\gamma^2 N^d}\right) \text{ for any constant } d.
\end{aligned}
$$

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.

First, note that the amount of flow traversing edges e during the first and last phase blocks of any constellation will be at most $\frac{p-1}{p-2}(g+1)r$. This is no more than 1 when $p \geq \frac{g+3}{\varepsilon} - 2$, which we conditioned on in the statement of the theorem.

Next, note that assuming no bad events occur, the amount of flow traversing edge e occurring during any other phase block of any constellation must be at most

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

□

Note that Theorem 1.3 is a direct corollary of Theorem 11.2.

4.4 Provably Separating the Capabilities Between ORNs and SORNs

In this section we show that semi-oblivious routing has a provable asymptotic advantage over oblivious routing in reconfigurable networks. In order to do so, we must compare the guaranteed throughput versus latency tradeoffs achieved by the family of SORN designs $\mathscr{S}_N(g, C)$ described above and distributions over ORN designs. We will show below that our family of SORN designs $\mathscr{S}_N(g, C)$ has a provable asymptotic advantage over ORNs in *average latency*. To do so, we provide the following lower bound on average (expected) latency of distributions over ORN designs.

Theorem 12. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $h = h(r) = \lfloor \frac{1}{2r} \rfloor$ and $\varepsilon_o = \varepsilon_o(r) = h + 1 - \frac{1}{2r}$, and let $L_{obl}(r, N)$ be the function*

$$L_{obl}(r, N) = \varepsilon_o(\varepsilon_o N)^{1/h} + N^{1/(h+1)}.$$

*Then for every $N > 1$ and every distribution of ORN designs $\mathscr{R}$ on N nodes that guarantees throughput r, the expected **average** latency of $\mathcal{R} \sim \mathscr{R}$ is at least $\Omega(L_{obl}(r, N))$.*

The proof of Theorem 12 follows a similar structure as the lower bound proof of [AWS+22], only with an added average latency constraint in the starting linear program, which results in an additional variable in the corresponding dual program, which must be reasoned about and assigned a value. We leave the proof to Appendix B.1.

Theorem 13. *Consider any constant $r \in (0, \frac{1}{2}]$, and let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$ Then if $r \in \left(0, \frac{1}{4}\right] \cup \left[\frac{1}{4-(2/N^{1/6})}, \frac{1}{3}\right]$ and $\frac{1}{r}$ is not an integer, the family of SORN designs $\mathscr{S}_N(g, C)$ achieves asymptotically better average latency than any family of ORN designs which guarantees throughput r.*

Proof. By Theorem 12, any family of ORN designs which guarantees throughput r must suffer average latency $\Omega\left(L_{obl}(r, N)\right)$. Also recall that the family of SORN designs $\mathscr{S}_N(g, C)$ achieves maximum latency $\tilde{O}(gN^{1/g})$ with high probability as long as $\frac{1}{r}$ is not an integer. This implies it also achieves average latency $\tilde{O}(gN^{1/g})$, since with probability 1 it achieves maximum latency $\tilde{O}(N)$. We divide the set of throughput values $r \in \left(0, \frac{1}{4}\right] \cup \left[\frac{1}{4-(2/N^{1/6})}, \frac{1}{3}\right]$ into the following cases.

1. $r \leq \frac{1}{5}$. Then $g(r) > h(r) + 1$. Since $L_{obl}(r, N) \geq N^{1/(h+1)}$, then $L_{obl}(r, N)$ is asymptotically greater than $\tilde{O}(L_{upp})$.
2. $r \in \left(\frac{1}{5}, \frac{1}{4}\right]$. Then $\varepsilon_o(r) = h + 1 - \frac{1}{2r} \geq \frac{1}{2}$, and $g(r) > h(r)$. Therefore, $L_{obl}(r, N)$ is asymptotically greater than $\tilde{O}(L_{upp})$.
3. $r \in \left[\frac{1}{4-(2/N^{1/6})}, \frac{1}{3}\right]$. Then $\varepsilon_o(r) \geq \frac{1}{N^{1/6}}$, $g(r) = 2$, and $h(r) = 1$. So $\varepsilon_o(\varepsilon_o N)^{1/h} \geq \left(\frac{1}{N^{1/6}}\right)^2 N = N^{2/3}$. Additionally, $gN^{1/g} = 2\sqrt{N}$. Therefore, $L_{obl}(r, N)$ is asymptotically greater than $\tilde{O}(L_{upp})$.

□

5 Lower Bound

In this section we prove Theorem 1.4, restated below.

Theorem 1.4. *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*

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

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

Proof. We will start by upper bounding throughput for a given maximum latency. We begin with a set of $N!$ linear programs, one for each possible permutation σ on the node set, to solve the following problem: given maximum latency L and some reconfigurable network schedule, LP_σ finds a set of routing paths to route r flow between each $a, \sigma(a)$ pair, and maximizes the value r for which this is possible.

Primal LP_σ

maximize	r	
subject to	$\sum_{P \in \mathcal{P}_L(a,\sigma(a),t)} \mathcal{R}_\sigma(a,t,P) = r$	$\forall a \in [N], t \in [T]$
	$\sum_{a,t} \sum_{P \in \mathcal{P}_L(a,\sigma(a),t): e \in P} \mathcal{R}_\sigma(a,t,P) \leq 1$	$\forall e \in E_{\text{phys}}$
	$\mathcal{R}_\sigma(a,t,P) \geq 0$	$\forall a \in [N], t \in [T], P \in \mathcal{P}_L(a,\sigma(a),t)$

We then take the dual program of each LP_σ to find Dual_σ.

Dual_σ

minimize	$\sum_e \beta_{\sigma e}$	
subject to	$\alpha_{at\sigma} \leq \sum_{e \in P} \beta_{\sigma e}$	$\forall a \in [N], t \in [T], P \in \mathcal{P}_L(a,\sigma(a),t)$
	$\sum_{at} \alpha_{at\sigma} \geq 1$	
	$\beta_{\sigma e} \geq 0$	$\forall e \in E_{\text{phys}}$

For each Dual_σ, we will define a dual solution. Then, we will analyze an upper bound on the objective value of Dual_σ, with high probability over the random sampling of σ.

We will also reframe Dual_σ in the following way, which will be easier to work with. Note that $\left(\sum \beta_{\sigma e}\right) / \left(\sum \alpha_{at\sigma}\right)$ is still an upper bound on throughput.

minimize	$\left(\sum_e \beta_{\sigma e}\right) / \left(\sum_{at} \alpha_{at\sigma}\right)$	
subject to	$\alpha_{at\sigma} \leq \sum_{e \in P} \beta_{\sigma e}$	$\forall a, t, P$
	$\beta_{\sigma e} \geq 0$	

To understand how we construct and analyze dual solutions for Dual_σ, we'll start by showing that oblivious designs cannot achieve throughput better than $1/2$, even with high probability. Define

$$\beta_{\sigma e} = \begin{cases} 2 & \text{if } e \text{ connects some } a \to \sigma(a) \text{ pair} \\ 1 & \text{otherwise.} \end{cases}$$

and let $\alpha_{at\sigma} = \min_{P \in \mathcal{P}_L(a,\sigma(a),t)}\{\sum_{e \in P} \beta_{\sigma e}\}$. By construction, $\alpha_{at\sigma} \geq 2$ for all a, t. So $\sum_{a,t} \alpha_{at\sigma} \geq 2NT$, where T is the period of the schedule.

Additionally, in expectation, $\mathbb{E}[\beta_{\sigma e}] = 1 + \frac{1}{N}$ for all e. So, $\mathbb{E}\left[\sum_e \beta_{\sigma e}\right] = (1 + \frac{1}{N})NT$
Then $\mathbb{E}[(\sum_e \beta_{\sigma e})/(\sum_{at} \alpha_{at\sigma})] \leq \frac{1}{2}\left(1 + \frac{1}{n}\right)$, which converges to $\frac{1}{2}$ as $N \to \infty$.

Now, suppose that throughput r is achievable with high probability. That would mean that routing the demands rD_σ gives a feasible flow with probability at least $\left(1 - \frac{1}{N}\right)$ over a uniformly random choice σ. If routing demands rD_σ is feasible for a fixed permutation σ, then it must be the case that the objective value of LP_σ is at least r.

And since the objective value of LP_σ is always non-negative, then this implies that over a uniformly random permutation σ, the expected objective value of LP_σ is at least $r \cdot (1 - 1/N)$.

The inequality $r \cdot (1 - 1/N) \leq \frac{1}{2}(1 + \frac{1}{N})$ implies that r must be at most $\frac{1}{2} + \frac{2}{N-1}$.

Dual_σ solutions to bound general throughput. Now we'll show good dual solutions for general r. Given parameter $\theta \in \mathbb{Z}_{\geq 1}$, set

$$\beta_{\tau e} = \begin{cases} \theta + 1 & \text{if } e \text{ on a path of } \theta \text{ phys edges between some } u \to \sigma(u) \text{ pair} \\ 1 & \text{otherwise.} \end{cases}$$

By construction, $\alpha_{at\sigma} \geq \theta + 1$, so $\sum_{a,t} \alpha_{at\sigma} = NT(\theta + 1)$, where T is the period. Additionally,

$$\mathbb{E}[\beta_{\sigma e}] = 1 + \theta \Pr(e \text{ is on a path of} \leq \theta \text{ phys edges with } \sigma\text{-matched endpoints})$$

To bound the above value, we apply the following lemma from [AWS+22].

Lemma 14. ***(Counting Lemma)[AWS+22]*** *If in an ORN topology, some node a can reach k other nodes in at most L timesteps using at most h physical hops per path for some integer h, then $k \leq 2\binom{L}{h}$, assuming $h \leq \frac{1}{3}L$.*

Applying the Counting Lemma, the probability that edge e is on a path of no more than θ physical edges with σ-matched endpoints is at most

$$\frac{1}{N}\sum_{m=0}^{\theta-1} 2\binom{L}{m}2\binom{L}{\theta - 1 - m} \leq \frac{4}{N}\binom{2L}{\theta - 1}$$

assuming $\theta - 1 \leq \frac{1}{3}L$. Then

$$\mathbb{E}[\beta_{\sigma e}] \leq 1 + \frac{4\theta}{N}\binom{2L}{\theta - 1}$$
$$\implies \mathbb{E}\left[\sum_e \beta \sigma e\right] \leq NT\left(1 + \frac{4\theta}{N}\binom{2L}{\theta - 1}\right)$$

Meaning we can bound the expected objective value of Dual_σ throughput achievable under random permutation traffic.

$$\begin{aligned} \mathbb{E}[\text{obj. value of Dual}_\sigma] &\leq \mathbb{E}\left[\sum_e \beta_{\sigma e}\right] / \left(NT(\theta + 1)\right) \\ &\leq \left(1 + \frac{4\theta}{N}\binom{2L}{\theta - 1}\right) / \left(\theta + 1\right) \end{aligned}$$

As before, we use this expectation to find an upper bound on the achievable throughput rate with high probability

$$r\left(1-\frac{1}{N^d}\right) \leq \left(1+\frac{4\theta}{N}\binom{2L}{\theta-1}\right)/\left(\theta+1\right)$$

We then simplify and isolate L to one side of the inequality, to find the following lower bound on maximum latency. The inequality $\frac{a!}{(a-b)!} \leq a^b$ and Stirling's approximation $(k!)^{\frac{1}{k}} \geq \frac{k}{e}\sqrt{2\pi k}^{\frac{1}{k}}$ prove useful during this simplification process.

$$L \geq \frac{\theta-1}{2e}N^{\frac{1}{\theta-1}}\left(\left(\frac{N^d-1}{N^d}r-\frac{1}{\theta+1}\right)\frac{\sqrt{2\pi(\theta-1)}}{4\theta}\right)^{\frac{1}{\theta-1}}$$

To ensure that this bound stays above 0, we approximately need $(r(\theta+1)-1)>0$, meaning θ must be greater than $\frac{1}{r}-1$. Setting θ as the smallest integer for which this holds, we find $\theta = \lfloor \frac{1}{r} \rfloor$. Let $g=\theta-1$ and $\varepsilon = g+1-(\frac{1}{r}-1)$. Then we substitute $r=\frac{1}{g+2-\varepsilon}$ to find

$$\begin{aligned} L &\geq \frac{g}{2e}\left(\varepsilon N\right)^{1/g}\left(\frac{\sqrt{2\pi g}}{4(h+1)(g+2-\varepsilon)}\right)^{1/g} \\ \implies L &\geq \Omega\left(g\left(\left(\varepsilon N\right)^{1/g}+N^{1/(g+1)}\right)\right). \end{aligned}$$

□

Corollary 15. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g=g(r)=\lfloor \frac{1}{r}-1 \rfloor$ and $\varepsilon = \varepsilon(r) = g+1-(\frac{1}{r}-1)$.*

1. *Then any fixed SORN design which guarantees throughput r (with respect to fixed demands), must suffer maximum latency at least $\Omega(L_{low}(r,N))$.*
2. *Additionally, any distribution over SORN designs $\mathscr{S}$ each of size N, which guarantees throughput r (with respect to fixed demands) over the random sampling $\mathcal{R} \sim \mathscr{R}$ must suffer at least $\Omega(L_{low}(r,N))$ maximum latency.*

Before we begin the proof, note that this lower bound does not make any claims about what maximum latencies are achievable with high probability for SORNs which guarantee throughput r. In Appendix B.2, we give a similar lower bound on the *average* (or, expected) latency of any SORN design which guarantees throughput r. This lower bound has an additional multiplicative dependence on ε. Thus, the lower bound on maximum latency and the lower bound on expected latency match to within a constant factor for most values of r: when $\frac{1}{r} \notin \bigcup_{m=2}^{\infty}\left(m-\frac{1}{K}, m\right)$, for any large constant K.

Proof. The linear program as written in the proof of Theorem 1.4, when considered as whole instead of as a family of $N!$ different programs, sets up this SORN problem exactly. It asks: given a particular reconfigurable network schedule, for each possible permutation σ, maximize the guaranteed throughput rate while routing flow between σ-matched pairs. Since the expectation $\mathbb{E}_\sigma\left[\sum_e \beta_{\sigma e}\right]$, is upper bound by $NT\left(1+\frac{4\theta}{N}\binom{2L}{\theta-1}\right)$, then there exists at least one σ for which the bound holds. The rest of the proof follows similarly to the proof of Theorem 1.4, only without the factor $\left(1-\frac{1}{N^d}\right)$.

Note that every SORN design $\mathcal{S}$ within the support of $\mathscr{S}$ must itself guarantee throughput r (with probability 1). Thus, each design $\mathcal{S}$ must suffer maximum latency at least $\Omega(L_{low}(r,N))$, and the whole distribution must also suffer maximum latency at least $\Omega(L_{low}(r,N))$. □

6 Conclusion and Open Questions

In this paper, we showed that, compared to the guaranteed throughput versus latency tradeoff achieved in [AWS+22], 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 $\mathcal{O}(L_{upp}(r,N))$ up to a $\tilde{\mathcal{O}}(\log N)$ factor, leaving a logarithmic gap between our upper and lower bounds. Is there an ORN or SORN design that achieves maximum latency $\mathcal{O}(L_{upp}(r,N))$? Alternatively, is there a stronger lower bound than the one we presented in Section 5?

Additionally, when $\varepsilon^{1/g}$ is sub-constant, then $L_{upp}(r,N) > \mathcal{O}(L_{low}(r,N))$. 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. [AWS+22] handled this case by developing a second ORN family which sent flow on both h- and $(h+1)$-hop semi-paths. We believe there a similar result for ORNs which achieve throughput with high probability and 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 $\mathscr{S}_N(g,C)$ 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 in this domain, we make several assumptions that do not hold for practical networks in order to make the analysis tractable. In particular, our model of ORNs does not account for propagation delay between nodes. In a practical network, 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 our 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. As a result, in real systems queuing may happen, 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.

References

[AAS23] Vamsi Addanki, Chen Avin, and Stefan Schmid. Mars: Near-optimal throughput with shallow buffers in reconfigurable datacenter networks. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–43, 2023.

[AC03] David L. Applegate and Edith Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In Anja Feldmann, Martina Zitterbart, Jon Crowcroft, and David Wetherall, editors, *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 25-29, 2003, Karlsruhe, Germany*, pages 313–324. ACM, 2003.

[ACF+03] Yossi Azar, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Räcke. Optimal oblivious routing in polynomial time. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 383–388, New York, NY, USA, 2003. Association for Computing Machinery.

[ALMN91] William A. Aiello, F. T. Leighton, Bruce M. Maggs, and Mark Newman. Fast algorithms for bit-serial routing on a hypercube. *Mathematical systems theory*, 24(1):253–271, 1991.

[AWS+22] Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, Robert Kleinberg, and Rachit Agarwal. Optimal oblivious reconfigurable networks. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 1339–1352, New York, NY, USA, 2022. Association for Computing Machinery.

[BC07] Moshe Babaioff and John Chuang. On the optimality and interconnection of valiant load-balancing networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 80–88. IEEE, 2007.

[BCB+20] Hitesh Ballani, Paolo Costa, Raphael Behrendt, Daniel Cletheroe, Istvan Haller, Krzysztof Jozwik, Fotini Karinou, Sophie Lange, Kai Shi, Benn Thomsen, et al. 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*, pages 782–797, 2020.

[BH85] Allan Borodin and John E. Hopcroft. Routing, merging, and sorting on parallel models of computation. *J. Comput. Syst. Sci.*, 30:130–145, 1985.

[BKR03] Marcin Bienkowski, Miroslaw Korzeniowski, and Harald Räcke. A practical algorithm for constructing oblivious routing schemes. In *Proceedings of the Fifteenth Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '03, page 24–33, New York, NY, USA, 2003. Association for Computing Machinery.

[BMB20] Kashinath Basu, Ali Maqousi, and Frank Ball. 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)*, pages 1–6. IEEE, 2020.

[CWW+14] Q. Cheng, A. Wonfor, J. L. Wei, R. V. Penty, and I. H. White. 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):5244–5247, Sep 2014.

[DR96] Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

[DWC+17] M. Ding, A. Wonfor, Q. Cheng, R. V. Penty, and I. H. White. Scalable, low-power-penalty nanosecond reconfigurable hybrid optical switches for data centre networks. In *2017 Conference on Lasers and Electro-Optics (CLEO)*, pages 1–2, May 2017.

[FPR+10] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *Proceedings of ACM SIGCOMM*, 2010.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[GHMP08] Albert Greenberg, James Hamilton, David A Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks, 2008.

[GHR06] Anupam Gupta, Mohammad Taghi Hajiaghayi, and Harald Räcke. Oblivious network design. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 970–979. ACM Press, 2006.

[GHZ21] Mohsen Ghaffari, Bernhard Haeupler, and Goran Zuzic. Hop-constrained oblivious routing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1208–1220, New York, NY, USA, 2021. Association for Computing Machinery.

[GMP+16] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel Kilper. Projector: Agile reconfigurable data center interconnect. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, page 216–229, New York, NY, USA, 2016. Association for Computing Machinery.

[GYG+17] Soudeh Ghorbani, Zibin Yang, P. Brighten Godfrey, Yashar Ganjali, and Amin Firoozshahian. Drill: Micro load balancing for low-latency data center networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, page 225–238, New York, NY, USA, 2017. Association for Computing Machinery.

[GZB+21] Chen Griner, Johannes Zerwas, Andreas Blenk, Manya Ghobadi, Stefan Schmid, and Chen Avin. 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):1–33, 2021.

[HHR03] Chris Harrelson, Kirsten Hildrum, and Satish Rao. A polynomial-time tree decomposition to minimize congestion. In Arnold L. Rosenberg and Friedhelm Meyer auf der Heide, editors, *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)*, pages 34–43. ACM, 2003.

[HKLR05] MohammadTaghi Hajiaghayi, Jeong Han Kim, Tom Leighton, and Harald Räcke. Oblivious routing in directed graphs with random demands. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 193–201, New York, NY, USA, 2005. Association for Computing Machinery.

[HQG+14] Navid Hamedazimi, Zafar Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, page 319–330, New York, NY, USA, 2014. Association for Computing Machinery.

[JDP83] Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.

[KCML05] Isaac Keslassy, Cheng-Shang Chang, Nick McKeown, and Duan-Shin Lee. Optimal load-balancing. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 1712–1722. IEEE, 2005.

[KKT91] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. Tight bounds for oblivious routing in the hypercube. *Math. Syst. Theory*, 24(4):223–232, 1991.

[KLS81] Alam Khursheed and KM Lai Saxena. Positive dependence in multivariate distributions. *Communications in Statistics-Theory and Methods*, 10(12):1183–1196, 1981.

[KYY+18] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiunlin Lim, and Robert Soulé. Semi-oblivious traffic engineering: The road not taken. In Sujata Banerjee and Srinivasan Seshan, editors, *15th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2018, Renton, WA, USA, April 9-11, 2018*, pages 157–170. USENIX Association, 2018.

[LLF+14] He Liu, Feng Lu, Alex Forencich, Rishi Kapoor, Malveeka Tewari, Geoffrey M. Voelker, George Papen, Alex C. Snoeren, and George Porter. Circuit switching under the radar with reactor. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 1–15, Seattle, WA, April 2014. USENIX Association.

[MDG+20] William M. Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C. Snoeren, and George Porter. Expanding across time to deliver bandwidth efficiency and low latency. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 1–18, Santa Clara, CA, February 2020. USENIX Association.

[NJ94] Ted Nesson and Lennart Johnsson. Romm routing: A class of efficient minimal routing algorithms. In Kevin Bolding and Lawrence Snyder, editors, *Parallel Computer Routing and Communication*, pages 185–199, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[NJ95] Ted Nesson and S. Lennart Johnsson. Romm routing on mesh and torus networks. In *Proceedings of the Seventh Annual ACM Symposium on Parallel Algorithms and Architectures*, SPAA '95, page 275–287, New York, NY, USA, 1995. Association for Computing Machinery.

[PSF+13] George Porter, Richard Strong, Nathan Farrington, Alex Forencich, Pang Chen-Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Integrating microsecond circuit switching into the data center. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, page 447–458, New York, NY, USA, 2013. Association for Computing Machinery.

[Räc02] H. Räcke. Minimizing congestion in general networks. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pages 43–52, 2002.

[Räc08] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. STOC '08, New York, NY, USA, 2008. Association for Computing Machinery.

[SVB+19] Vishal Shrivastav, Asaf Valadarsky, Hitesh Ballani, Paolo Costa, Ki Suh Lee, Han Wang, Rachit Agarwal, and Hakim Weatherspoon. Shoal: A network architecture for disaggregated racks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, Boston, MA, 2019. USENIX Association.

[Val82] Leslie G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350–361, 1982.

[Val83] Valiant. Optimality of a two-phase strategy for routing in interconnection networks. *IEEE Transactions on Computers*, C-32(9):861–863, 1983.

[VB81] Leslie G. Valiant and Gordon J. Brebner. Universal schemes for parallel communication. pages 263–277, 1981.

[Waj17] David Wajc. Lecture notes: Negative association - definition, properties, and applications, April 2017.

[WAK+10] Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T.S. Eugene Ng, Michael Kozuch, and Michael Ryan. C-through: Part-time optics in data centers. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, page 327–338, New York, NY, USA, 2010. Association for Computing Machinery.

[WAS+23] Tegan Wilson, Daniel Amir, Vishal Shrivastav, Hakim Weatherspoon, and Robert Kleinberg. Extending optimal oblivious reconfigurable networks to all n. In *Proceedings of the SIAM Symposium on Algorithmic Principles of Computer Systems (APOCS)*, 2023.

[ZHR23] Goran Zuzic, Bernhard Haeupler, and Antti Roeyskoe. Sparse semi-oblivious routing: Few random paths suffice. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing*, PODC '23, page 222–232, New York, NY, USA, 2023. Association for Computing Machinery.

[ZSM05] Rui Zhang-Shen and Nick McKeown. 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, pages 178–192. Springer, 2005.

[ZZZ+12] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 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*, SIGCOMM '12, page 443–454, New York, NY, USA, 2012. Association for Computing Machinery.

A Mixing $(g+1)$-hop and 2-hop paths in our Semi-Oblivious Design

In defining the SORN design $\mathcal{S}$, we always chose to route permutation demand D_σ on 2-hop paths if any edge e in any (C, g)-constellation would become overloaded from routing D_σ on $(g+1)$-hop paths. However, this choice is a bit extreme. After all, the connection schedule of $\mathcal{S}$ iterates through $(p-1)^{g-1}$ different (C, g)-constellations.

Label a (C, g)-constellation $A\mathcal{V}$ as *contentious* if there exists some edge e occurring during constellation $A\mathcal{V}$ which is overloaded when routing demand D_σ with the $(g+1)$-hop routing scheme. It would be desirable the flow which would be routed along non-contentious constellations could still be routed on the more latency-efficient $(g+1)$-hop paths, while only the flow that would be routed on the contentious constellations is relegated to the 2-hop alternate paths.

This strategy slightly decreases the achievable throughput rate, due to reserving a small amount of edge capacity on each edge for 2-hop paths. However, as long as the number of contentious (C, g)-constellations k is small, we can still provably achieve throughput r for any $r \in (0, \frac{1}{2})$ for which $\varepsilon(r) = \left\lfloor \frac{1}{r} - 1 \right\rfloor + 1 - (\frac{1}{r} - 1) \neq 1$. (Or in other words, for r which is not the reciprocal of an integer.)

Specifically, if there are no more than $\frac{(1-\varepsilon)(p-2)^g}{4(p-1)}$ contentious (C, g)-constellations over the entire period, then as described above, only route the flow that would be routed on contentious constellations on the alternate 2-hop paths. (This is exactly the flow that originates during a constellation immediately prior to a contentious constellation.) Route all other flow on $(g+1)$-hop paths. If there are more than $\frac{(1-\varepsilon)(p-2)^g}{4(p-1)}$ contentious (C, g)-constellations, then route all flow on 2-hop paths.

Corollary 16. *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 $\varepsilon \neq 1$. Let $\delta = \frac{1-\varepsilon}{2(g-1)}$ and $C = \frac{6 \log\log N}{\delta^2} \ln(N)$, and assume that $N = p^g$ for prime p for which $C(g+1) \leq p$. Consider the SORN design $\mathcal{S}$ described above with parameters C and g, with the following alteration.*

Then this scheme can guarantee throughput r and achieves maximum latency $\tilde{\boldsymbol{O}}\left(gN^{1/g}\right)$ with high probability over the random sampling over σ, and achieves maximum latency $\tilde{\boldsymbol{O}}(N)$ in the low-probability case.

Proof. Suppose that k different (C, g)-constellations are contentious, and thus the flow which we would like to send only on $(g+1)$-hop paths within those frames must instead be sent on 2-hop paths across two iterations of the schedule. This presents a balancing problem: since most (C, g)-constellations are not contentious, most of the edges this flow will be sent on have their own constellation's $(g+1)$-hop flows to forward along. Thus, we need to bound the total amount of 2-hop flow on any edge in the network, given that k different frame's worth of flow is being routed on 2-hop paths.

Fix an edge e, and consider for each demand pair $i, \sigma(i)$ how much much flow is crossing edge e due to $i, \sigma(i)$. If 1st hop flow crosses edge e from i to $\sigma(i)$, then it must be the case that $tail(e) = i$ and both $head(e) - i$ and $\sigma(i) - head(e)$ have only non-zero coordinates. Similarly, if 2nd hop flow crosses edge e from i to $\sigma(i)$, then $head(e) = \sigma(i)$ and both $tail(e) - i$ and $\sigma(i) - tail(e)$ have only non-zero coordinates.

If there are k contentious (C, g)-constellations, then the total amount of flow that must be routed on 2-hop paths over the entire period will be $rkNC(g+1)(p-1)$, with each demand pair $i, \sigma(i)$ contributing $rkC(g+1)(p-1)$ flow per period.

For each edge $e = (a, b)$, consider the total amount of first-hop flow from 2-hop paths traversing the edge. First-hop flow traversing e must be traveling from source node $i = a$. Also note that

since edge e exists in the network, then the vector $b - a$ must have only non-zero coordinates. Then first-hop flow traverses edge e only when $\sigma(a) - b$ also has only non-zero coordinates.

For node a, let us consider the set of other 2-hop paths which could carry flow from a to $\sigma(a)$. (And thus, what other edges could carry first-hop 2-hop flow from a to $\sigma(a)$.) This is directly related to the number of nodes b' for which $\sigma(a) - b'$ and $b' - a$ both have non-zero coordinates. This is at least $(p-2)^g$, which occurs exactly when a and $\sigma(a)$ have no matching coordinates. Additionally, for a given first-hop edge e, the number of times an equivalent edge appears at any point in the period is the number of Vandermonde vectors in the constellation, or $C(g+1)$.

Thus, the amount of first-hop 2-hop flow that traverses edge e is always no more than

$$\frac{rkC(g+1)(p-1)}{(p-2)^g C(g+1)} = \frac{rk(p-1)}{(p-2)^g}.$$

A similar argument shows that the amount of second-hop 2-hop flow traversing edge e will also be no more than $\frac{k(p-1)}{(p-2)^g}$.

Now that we have this bound, let us bound the total amount of $(g+1)$-hop and 2-hop flow traversing some edge e.

Fix an edge e from a constellation that is not contentious. This edge will have both $(g+1)$-hop and 2-hop flow traversing it. Since the constellation is not contentious, we know that the amount of $(g+1)$-hop flow traversing e is no more than $(1+\delta)(g+1)r$. Thus, the total amount of flow traversing edge e is no more than

$$(1+\delta)(g+1)r + \frac{2rk(p-1)}{(p-2)^g}$$

Setting this value equal to 1, thus maximizing r, we achieve

$$\begin{aligned}(1+\delta)(g+1)r + \frac{2rk(p-1)}{(p-2)^g} &= 1\\ r\left((1+\delta)(g+1) + \frac{2k(p-1)}{(p-2)^g}\right) &= 1\end{aligned}$$

Now replace $\delta = \frac{1-\varepsilon}{2(g+1)}$ and $r = \frac{1}{g+2-\varepsilon}$ and solve for k to find the maximum value k may take without overloading any edges.

$$\begin{aligned}\frac{1}{g+2-\varepsilon}\left(\left(1+\frac{1-\varepsilon}{2(g+1)}\right)(g+1) + \frac{2k(p-1)}{(p-2)^g}\right) &= 1\\ \left(1+\frac{1-\varepsilon}{2(g+1)}\right)(g+1) + \frac{2k(p-1)}{(p-2)^g} &= g+2-\varepsilon\\ g+1+\frac{1-\varepsilon}{2} + \frac{2k(p-1)}{(p-2)^g} &= g+2-\varepsilon\\ \frac{2k(p-1)}{(p-2)^g} &= \frac{1-\varepsilon}{2}\\ k &= \frac{(1-\varepsilon)(p-2)^g}{4(p-1)}\end{aligned}$$

As stated in the theorem statement, this is the maximum value k can take without overloading edges in the network.

Now consider the probability that k (C, g)-constellations are contentious. This is clearly no more than the probability that a single (C, g)-constellation is contentious, which occurs with high probability as stated in the proof of Theorem 11.

□

B Average Latency Lower Bounds

B.1 Oblivious Designs

We devote this section to proving Theorem 12 as stated in Section 4.4, restated below.

Theorem 12. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $h = h(r) = \lfloor \frac{1}{2r} \rfloor$ and $\varepsilon_o = \varepsilon_o(r) = h + 1 - \frac{1}{2r}$, and let $L_{obl}(r, N)$ be the function*

$$L_{obl}(r, N) = \varepsilon_o(\varepsilon_o N)^{1/h} + N^{1/(h+1)}.$$

*Then for every $N > 1$ and every distribution of ORN designs $\mathscr{R}$ on N nodes that guarantees throughput r, the expected **average** latency of $\mathcal{R} \sim \mathscr{R}$ is at least $\Omega(L_{obl}(r, N))$.*

Proof. We begin by showing that the average latency of any fixed ORN design which guarantees throughput r with respect to time-stationary demands must satisfy average latency at least $\Omega(L_{obl}(r, N))$. This will be enough to proof Theorem 12. Note that every ORN design $\mathcal{R}$ within the support of $\mathscr{R}$ must guarantee throughput r (with probability 1). Thus, each design $\mathcal{R}$ must satisfy average latency at least $\Omega(L_{obl}(r, N))$. Use linearity of expectation to then show that the expected average latency of $\mathcal{R} \sim \mathscr{R}$ must be at least $\Omega(L_{obl}(r, N))$.

Fix any ORN connection schedule $\boldsymbol{\pi}$. We begin by stating the following linear program which, given $\boldsymbol{\pi}$ and average latency bound L, attempts to find a routing scheme which maximizes throughput, while keeping the average latency among all routing paths used, weighted by the fraction of flow traveling along each path, below the average latency bound L.

The proof will continue in the following way: we will first transform our LP into another LP which has fewer constraints. Then, we will take the Dual, to turn it into a minimization problem. We will give a dual solution and upper bound its objective value, thus upper bounding guaranteed throughput subject to an average latency constraint. Finally, we will rewrite this inequality into a lower bound on average latency, subject to a guaranteed throughput.

Primal LP

maximize	r	
subject to	$\sum_{P \in \mathcal{P}(a,b,t)} \mathcal{R}_{a,b,t}(P) = r$	$\forall a, b \in [N], t \in [T]$
	$\sum_{a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t) : e \in P} \mathcal{R}_{a,\sigma(a),t}(P) \leq 1$	$\forall e \in E_{\text{phys}}, \sigma$ permut on $[N]$
	$\sum_{a,b,t} \sum_{P \in \mathcal{P}(a,b,t)} \mathcal{R}_{a,b,t} \cdot lat(P) \leq rN^2T \cdot L$	
	$\mathcal{R}_{a,b,t}(P), r \geq 0$	$\forall a, b \in [N], t \in [T], P \in \boldsymbol{\mathcal{P}}(a, b, t)$

Where we interpret $lat(P)$ as the latency of the path P, or the combined number of virtual and physical edges[5]. As in [AWS+22], we replace the factorial number of constraints ranging over choices of σ with a polynomial number of constraints which range over choices of $a, b \in [N]$. We do this by interpreting these constraints for a fixed edge e as solving a maximum bipartite matching problem from $[N]$ to $[N]$. See Section 3.1 of [AWS+22] for a step-by-step explanation.

[5]We use $lat(P)$ here instead of $L(P)$ as in Section 2 to denote the latency of path P to prevent confusion between the latency of a path and the average latency bound L.

Primal LP

maximize	r	
subject to	$\sum_{P \in \mathcal{P}(a,b,t)} \mathcal{R}_{a,b,t}(P) = r$	$\forall a, b \in [N], t \in [T]$
	$\sum_a \xi_{a,e} + \sum_b \eta_{b,e} \leq 1$	$\forall e \in E_{\text{phys}}$
	$\sum_t \sum_{P \in \mathcal{P}(a,\sigma(a),t): e \in P} \mathcal{R}_{a,\sigma(a),t}(P) \leq \xi_{a,e} + \eta_{b,e}$	$\forall a, b \in [N], e \in E_{\text{phys}}$
	$\sum_{a,b,t} \sum_{P \in \mathcal{P}(a,b,t)} \mathcal{R}_{a,b,t} \cdot lat(P) \leq r \cdot N^2 TL$	
	$\mathcal{R}_{a,b,t}(P), \xi_{a,e}, \eta_{b,e}, r \geq 0$	$\forall a, b \in [N], t \in [T], P \in \mathcal{P}(a,b,t), e \in E_{\text{phys}}$

Dual

minimize	$\sum_e z_e$	
subject to	$\sum_{a,b,t} x_{a,b,t} - \gamma \cdot N^2 TL \geq 1$	
	$z_e \geq \sum_b y_{a,b,e}$	$\forall a \in [N], e \in E_{\text{phys}}$
	$z_e \geq \sum_a y_{a,b,e}$	$\forall b \in [N], e \in E_{\text{phys}}$
	$\sum_{e \in P} y_{a,b,e} + \gamma \cdot lat(P) \geq x_{a,b,t}$	$\forall a, b \in [N],\ t \in [T],\ P \in \mathcal{P}(a,b,t)$
	$y_{a,b,e}, z_e, \gamma \geq 0$	$\forall a, b \in [N],\ e \in E_{\text{phys}}$

We will first create a dual solution, aiming to fulfill all constraints except the first. We will then normalize the variables so that $\sum_{a,b,t} x_{a,b,t} - \gamma \cdot N^2 TL$ is as close to 1 as possible.

We define this value $m_\theta^+(e,a)$ as follows.

$$m_\theta^+(e,a) = \begin{cases} 1 & \text{if } e \text{ can be reached from } a \text{ using at most } \theta \text{ physical hops} \\ & \quad \text{(including } e\text{) in } \leq kL \text{ timesteps} \\ 0 & \text{if otherwise} \end{cases}$$

We define a similar value for edges which can reach node b.

$$m_\theta^-(e,b) = \begin{cases} 1 & \text{if } b \text{ can be reached from } e \text{ using at most } \theta \text{ physical hops} \\ & \quad \text{(including } e\text{) in } \leq kL \text{ timesteps} \\ 0 & \text{if otherwise} \end{cases}$$

Set $\hat{y}_{a,b,e} = m_\theta^+(e,a) + m_\theta^-(e,b)$. Also set $\hat{\gamma} = \frac{2\theta}{kL}$, and set $\hat{x}_{a,b,t} = \min_{P \in \mathcal{P}(a,b,t)} \{\sum_{e \in P} \hat{y}_{a,b,e} + \hat{\gamma} \cdot lat(P)\}$. Note that by definition, $\hat{\gamma}, \hat{x}$ and $\hat{y}$ variables satisfy the last set of dual constraints.

Consider some path P which connects a to b starting at timestep t. If path P has latency greater than kL, then

$$\sum_{e \in P} \hat{y}_{a,b,e} + \hat{\gamma} \cdot lat(P) \geq \hat{\gamma} kL = 2\theta.$$

If on the other hand, path P has latency no more than kL but uses at least θ physical hops, then

$$\sum_{e \in P} \hat{y}_{a,b,e} + \hat{\gamma} \cdot lat(P) \geq \sum_{e \in P} \hat{y}_{a,b,e} \geq 2\theta.$$

Finally, if path P has latency no more than kL and uses fewer than θ physical hops, then

$$\sum_{e \in P} \hat{y}_{a,b,e} + \hat{\gamma} \cdot lat(P) \geq \sum_{e \in P} \hat{y}_{a,b,e} = 2|P \cap E_{\text{phys}}|.$$

We use the following lemma, restated from Section 5, to bound $\sum_{a,b,t} \hat{x}_{abt}$.

Lemma 17. ***(Counting Lemma)[AWS+22]*** *If in an ORN topology, some node a can reach k other nodes in at most L timesteps using at most h physical hops per path for some integer h, then $k \leq 2\binom{L}{h}$, assuming $h \leq \frac{1}{3}L$.*

$$\begin{aligned}
\sum_{a,b,t} \hat{x}_{a,b,t} &\geq \sum_{a,t} \sum_{b \neq a} \min\{2\theta, \min_{P \in \mathcal{P}_{kL}(a,b,t)} \{2|P \cap E_{\text{phys}}|\}\} \\
&\geq NT\left(2\theta\left(N - 2\binom{kL}{\theta - 1}\right) + 2\binom{kL}{\theta - 1}\right) \\
\implies \sum_{a,b,t} \hat{x}_{a,b,t} - \hat{\gamma} N^2 TL &\geq NT\left(2\theta\left(N - 2\binom{kL}{\theta - 1}\right) + 2\binom{kL}{\theta - 1}\right) - \frac{2\theta}{k} N^2 T \\
&= NT\left(\left(2\theta - \frac{2\theta}{k}\right) N - 4\theta\binom{kL}{\theta - 1} + 4\binom{kL}{\theta - 1}\right) = w
\end{aligned}$$

Set this equal to w, our normalization term for each of the dual variables. Now set $\gamma = \frac{1}{w}\hat{\gamma}$, $y_{a,b,e} = \frac{1}{w}\hat{y}_{a,b,e}$ and $x_{a,b,t} = \frac{1}{w}\hat{x}_{a,b,t}$.

Finally, set $z_e = \max_{a,b}\{\sum_a y_{a,b,e}, \sum_b y_{a,b,e}\}$. Note that by construction, our dual solution satisfies all constraints. To bounds throughput from above, we upper bound the sums $\sum_a y_{a,b,e}$ and $\sum_b y_{a,b,e}$, allowing us to upper bound the total sum of z_e variables.

$$\sum_a y_{a,b,e} = \frac{1}{w} \sum_a \left(m_\theta^+(e,a) + m_\theta^-(e,b)\right) \leq \frac{1}{w}\left(\sum_a m_\theta^+(e,a) + N - 1\right) \leq \frac{1}{w}\left(2\binom{L}{\theta - 1} + N - 1\right)$$

where the last step is an application of the Counting Lemma. Similarly,

$$\sum_b y_{a,b,e} = \frac{1}{w} \sum_b \left(m_\theta^+(e,a) + m_\theta^-(e,b)\right) \leq \frac{1}{w}\left(N - 1 + \sum_b m_\theta^-(e,b)\right) \leq \frac{1}{w}\left(N - 1 + 2\binom{L}{\theta - 1}\right)$$

Recalling that $z_e = \max_{a,b}\{\sum_a y_{a,b,e}, \sum_b y_{a,b,e}\}$, and that the dual objective aims to minimize $\sum_e z_e$, we deduce that

$$
\begin{aligned}
r \leq \sum_e z_e &\leq \frac{NT}{w}\left(N-1+2\binom{kL}{\theta-1}\right) \\
&= \frac{N-1+2\binom{L}{\theta-1}}{\left(2\theta-\frac{2\theta}{k}\right)N-4\theta\binom{kL}{\theta-1}+4\binom{kL}{\theta-1}} \\
&\leq \frac{N-1+2\frac{(kL)!}{(\theta-1)!(kL-\theta+1)!}}{2\theta\left(\left(\frac{k-1}{k}\right)N-2\frac{(kL)!}{(\theta-1)!(kL-\theta+1)!}\right)} \\
&= \frac{k}{2\theta(k-1)}+\frac{4(kL)!}{2\theta(kL-\theta+1)!\left(\left(\frac{k-1}{k}\right)N(\theta-1)!-2\frac{(kL)!}{(kL-\theta+1)!}\right)} \\
&\leq \frac{k}{2\theta(k-1)}+\frac{4(kL)^{\theta-1}}{2\theta\left(\left(\frac{k-1}{k}\right)N(\theta-1)!-2(kL)^{\theta-1}\right)}
\end{aligned}
$$

using the fact that $\frac{a!}{(a-b)!} \leq a^b$. At this point, we rearrange the inequality to isolate L.

$$
\begin{aligned}
kL &\geq \left(\frac{\left(r-\frac{k}{2\theta(k-1)}\right)2\theta\frac{k-1}{k}N(\theta-1)!}{4\left(1+\theta\left(r-\frac{k}{2\theta(k-1)}\right)\right)}\right)^{\frac{1}{\theta-1}} \\
L &\geq \frac{\theta-1}{ke}N^{\frac{1}{\theta-1}}\left(\frac{\left(r-\frac{k}{2\theta(k-1)}\right)2\theta\frac{k-1}{k}\sqrt{2\pi(\theta-1)}}{4\left(1+\theta\left(r-\frac{k}{2\theta(k-1)}\right)\right)}\right)^{\frac{1}{\theta-1}}
\end{aligned}
$$

using Stirling's approximation, in the form $(k!)^{\frac{1}{k}} \geq \frac{k}{e}\sqrt{2\pi k}^{\frac{1}{k}}$.

Recall that $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon_o = h+1-\frac{1}{2r}$, as in the statement of the theorem above. We also set the parameter $\theta = h+1$. Note that our lower bound will always be positive when $\left(r-\frac{k}{2\theta(k-1)}\right) > 0$, which occurs as long as $\varepsilon_o > \frac{h+1}{k}$. This tells us how to set the constant k: we may set $k = 2\frac{h+1}{\varepsilon_o}$. Since $\varepsilon_o \in (0,1]$, this is always well-defined. Substitute h, ε_o into the lower bound and simplify.

$$
\begin{aligned}
L &\geq \frac{h}{ke} N^{\frac{1}{h}} \left(\frac{\left(r - \frac{k}{2(h+1)(k-1)}\right) 2(h+1)\frac{k-1}{k}\sqrt{\pi h/2}}{1 + (h+1)\left(r - \frac{k}{2(h+1)(k-1)}\right)} \right)^{\frac{1}{h}} \\
&= \frac{h}{ke} N^{\frac{1}{h}} \left(\sqrt{\frac{h\pi}{2}} \cdot \frac{\left(\frac{1}{2(h+1-\varepsilon_o)} - \frac{k}{2(h+1)(k-1)}\right)(h+1)\frac{k-1}{k}}{1 + (h+1)\left(\frac{1}{2(h+1-\varepsilon_o)} - \frac{k}{2(h+1)(k-1)}\right)} \right)^{\frac{1}{h}} \\
&= \frac{h}{ke} N^{\frac{1}{h}} \left(\sqrt{\frac{h\pi}{2}} \cdot \frac{k-1}{k} \cdot \frac{k\varepsilon_o - (h+1)}{3(h+1)(k-1) - 2\varepsilon_o(k-1)} \right)^{\frac{1}{h}} \\
&= \frac{h}{ke} (\varepsilon_o N)^{\frac{1}{h}} \left(\sqrt{\frac{h\pi}{2}} \cdot \frac{k-1}{k} \cdot \frac{k - \frac{h+1}{\varepsilon_o}}{3(h+1)(k-1) - 2\varepsilon_o(k-1)} \right)^{\frac{1}{h}} \\
&\geq \frac{h}{ke} (\varepsilon_o N)^{\frac{1}{h}} \left(\sqrt{\frac{h\pi}{2}} \cdot \frac{1}{3(h+1) - 2\varepsilon_o} \right)^{\frac{1}{h}} \\
&\geq \Omega\left(\frac{h}{k} (\varepsilon_o N)^{\frac{1}{h}} \right) = \Omega\left(\varepsilon_o (\varepsilon_o N)^{\frac{1}{h}} \right)
\end{aligned}
$$

because $\frac{1}{k} = \frac{\varepsilon_o}{2(h+1)}$. Finally, we realize that any lower bound on average latency subject to a guaranteed throughput constraint $r' < r$ is also a lower bound on average latency subject to guaranteed throughput r. Let $r' = \frac{1}{2(h+1)}$. Then $r' < r$. Additionally,

$$
\Omega\left(\varepsilon_o(r') (\varepsilon_o(r') N)^{\frac{1}{h(r')}} \right) = \Omega\left(N^{\frac{1}{h+1}} \right).
$$

Therefore, combining these two lower bounds, we find that average latency of an ORN design which guarantees throughput r must be at least

$$
\Omega\left(\varepsilon_o (\varepsilon_o N)^{\frac{1}{h}} + N^{\frac{1}{h+1}} \right) = \Omega\left(L_{obl}(r, N) \right).
$$

□

B.2 Semi-Oblivious Designs

Theorem 18. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $g = g(r) = \left\lfloor \frac{1}{r} - 1 \right\rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and let $L_{sem}(r, N)$ be the function*

$$
L_{sem}(r, N) = \varepsilon(\varepsilon N)^{1/g} + N^{1/(g+1)}.
$$

Then for every $N > 1$ and every ORN design on N nodes that achieves throughput r with high probability, the average latency suffered by routing paths must be at least $\Omega(L_{sem}(r, N))$.

Proof. We start by upper bounding throughput for a given average latency bound. We begin with a set of $N!$ linear programs, one for each possible permutation σ on the node set, to solve the following problem: given an average latency bound L and some reconfigurable network schedule, LP_σ finds a set of routing paths to route r flow between each $a, \sigma(a)$ pair, and maximizes the value r for which this is possible.

Primal LP

maximize	r	
subject to	$\sum_{P \in \mathcal{P}(a,\sigma(a),t)} S_\sigma(a,t,P) = r$	$\forall a \in [N], t \in [T], \sigma$ permut on $[N]$
	$\sum_{a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t): e \in P} S_\sigma(a,t,P) \leq 1$	$\forall e \in E_{\text{phys}}, \sigma$ permut
	$\sum_{a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t): e \in P} S_\sigma(a,t,P) \cdot lat(P) \leq r \cdot NTLN!$	
	$S_\sigma(a,t,P) \geq 0$	$\forall a \in [N], t \in [T], \sigma$ permut, $P \in \mathcal{P}(a,\sigma(a),t)$

We then take the dual program of each LP to find the Dual program.

Dual$_\sigma$

minimize	$\sum_{e,\sigma} \beta_{\sigma e}$	
subject to	$\sum_{a,t,\sigma} \alpha_{at\sigma} - \gamma NTLN! \geq 1$	
	$\alpha_{at\sigma} \leq \sum_{e \in P} \beta_{\sigma e} + \gamma \cdot lat(P)$	$\forall a \in [N], t \in [T], \sigma$ permut, $P \in \mathcal{P}(a,\sigma(a),t)$
	$\gamma, \beta_{\sigma e} \geq 0$	$\forall e \in E_{\text{phys}}$

For each permutation σ, we will define it's associated Dual variables. Then, we will analyze an upper bound on the objective value of the entire Dual program.

We will also reframe the Dual program in the following way, which will be easier to work with. Note that $(\sum \beta_{\sigma e}) \,/\, (\sum \alpha_{at\sigma} - \gamma NTLN!)$ is still an upper bound on throughput.

minimize	$\left(\sum_{e,\sigma} \beta_{\sigma e}\right) / \left(\sum_{a,t,\sigma} \alpha_{at\sigma} - \gamma NTLN!\right)$	
subject to	$\alpha_{at\sigma} \leq \sum_{e \in P} \beta_{\sigma e} + \gamma \cdot lat(P)$	$\forall a, t, \sigma, P$
	$\gamma, \beta_{\sigma e} \geq 0$	

Given parameter $\theta \in \mathbb{Z}_{\geq 1}$, set

$$\beta_{\sigma e} = \begin{cases} \theta + 1 & \text{if } e \text{ on a path of } \leq \theta \text{ phys edges and } \leq kL \text{ latency} \\ & \quad \text{between some } u \to \sigma(u) \text{ pair} \\ 1 & \text{otherwise.} \end{cases}$$

And set $\gamma = \frac{\theta+1}{kL}$. Then for any path P, $\sum_{e \in P} \beta_{\sigma e} + \gamma \cdot lat(P) \geq \theta + 1$. Therefore, we can always assign $\alpha_{at\sigma} = \theta + 1$, giving us

$$\sum_{a,t,\sigma} \alpha_{at\sigma} - \gamma NTLN! = (\theta+1)NTN! \left(1 - \frac{1}{k}\right)$$

Finally, we upper bound $\mathbb{E}_\sigma[\sum_e \beta_{\sigma e}]$ to achieve an upper bound on $\sum_{e,\sigma} \beta_{\sigma e}$. We do this by upper bounding the expected value of the individual terms $\beta_{\sigma e}$.

$$\mathbb{E}_\sigma[\beta_{\sigma e}] = 1 + \theta \Pr[e \text{ is on a path of } \leq \theta \text{ phys edges and } \leq kL \text{ lat. with } \sigma\text{-matched endpoints}]$$

Applying the Counting Lemma (thus assuming $\theta - 1 \leq \frac{1}{3}L$), the above probability is at most

$$\frac{1}{N}\sum_{m=0}^{\theta-1} 2\binom{kL}{m}2\binom{kL}{\theta-1-m} \leq \frac{4}{N}\binom{2kL}{\theta-1}$$

This is a sum over the number of physical hops m taken before edge e. For each value m, we multiply the number of nodes a which can reach edge e using m physical hops in latency no more than kL by $\frac{1}{N}$ times the number of nodes b reachable from e using the remaining $(\theta-1-m)$ physical hops in latency no more than kL. Then

$$\mathbb{E}_\sigma[\beta_{\sigma e}] \leq 1 + \frac{4\theta}{N}\binom{2kL}{\theta-1}$$
$$\implies \mathbb{E}_\sigma\left[\sum_e \beta\sigma e\right] \leq NT\left(1 + \frac{4\theta}{N}\binom{2kL}{\theta-1}\right)$$

Meaning we can bound the expected objective value of Dual_σ throughput achievable under random permutation traffic.

$$\begin{aligned}\mathbb{E}[\text{obj. value of Dual}] &\leq \mathbb{E}_\sigma\left[\sum_e \beta_{\sigma e}\right] \Big/ \left(NT(\theta+1)\left(\frac{k-1}{k}\right)\right) \\ &\leq k\left(1 + \frac{4\theta}{N}\binom{2kL}{\theta-1}\right) \Big/ (\theta+1)(k-1)\end{aligned}$$

Therefore, the guaranteed throughput rate of any SORN design must be

$$r \leq k\left(1 + \frac{4\theta}{N}\binom{2kL}{\theta-1}\right) \Big/ (\theta+1)(k-1)$$

We then simplify and isolate L to one side of the inequality, to find the following lower bound on maximum latency. The inequality $\frac{a!}{(a-b)!} \leq a^b$ and Stirling's approximation $(a!)^{\frac{1}{a}} \geq \frac{a}{e}\sqrt{2\pi a}^{\frac{1}{a}}$ prove useful during this simplification process.

$$L \geq \frac{\theta-1}{2ke}N^{\frac{1}{\theta-1}}\left(\sqrt{2\pi(\theta-1)}\frac{\frac{r(\theta+1)(k-1)}{k}-1}{4\theta}\right)^{\frac{1}{\theta-1}}$$

To ensure the bound is positive, we need $\frac{r(\theta+1)(k-1)}{k} - 1 > 0$, meaning that we need for $\theta > \frac{k}{(k-1)r} - 1$, or approximately $\theta > \frac{1}{r} - 1$. Setting θ as the smallest integer for which this holds, we find $\theta = \lfloor\frac{1}{r}\rfloor$. Recall that $g = g(r) = \lfloor\frac{1}{r}\rfloor - 1$, therefore $\theta = g + 1$. Additionally recall that $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$. We substitute $r = \frac{1}{g+2-\varepsilon}$ and set $k = 2\frac{g+2}{\varepsilon}$. Thus, the factor $\frac{1}{k}$ becomes $\frac{\varepsilon}{2(g+2)}$, allowing the following lower bound on average latency to hold.

$$L \geq \frac{g}{2ke}(\varepsilon N)^{1/g}\left(\sqrt{\frac{\pi g}{2}}\cdot\frac{k - \frac{g+2}{\varepsilon}}{k(g+2-\varepsilon)(g+1)}\right)^{1/g}$$
$$\implies L \geq \Omega\left(\varepsilon(\varepsilon N)^{1/g}\right).$$

Finally, we realize that any lower bound on average latency subject to a guaranteed throughput constraint $r' < r$ is also a lower bound on average latency subject to guaranteed throughput r. Let $r' = \frac{1}{g+2}$. Then $r' < r$. Additionally,

$$\Omega\left(\varepsilon(r')(\varepsilon(r')N)^{\frac{1}{g(r')}}\right) = \Omega\left(N^{\frac{1}{g+1}}\right).$$

Therefore, combining these two lower bounds, we find that average latency of an ORN design which guarantees throughput r must be at least

$$\Omega\left(\varepsilon(\varepsilon N)^{\frac{1}{g}} + N^{\frac{1}{g+1}}\right) = \Omega\left(L_{sem}(r, N)\right).$$

□