

Universal Connection Schedules for Reconfigurable Networking*

Shaleen Baral[†]

Robert Kleinberg[†]

Sylvan Martin[†]

Henry Rogers[†]

Tegan Wilson[‡]

Ruogu Zhang^{*}

Abstract

Reconfigurable networks are a novel communication paradigm in which the pattern of connectivity between hosts varies rapidly over time. Prior theoretical work explored the inherent tradeoffs between throughput (or, hop-count) and latency, and showed the existence of infinitely many Pareto-optimal designs as the network size tends to infinity. Existing Pareto-optimal designs use a connection schedule which is fine-tuned to the desired hop-count h , permitting lower latency as h increases. However, in reality datacenter workloads contain a mix of low-latency and high-latency requests. Using a connection schedule fine-tuned for one request type leads to inefficiencies when serving other types.

A more flexible and efficient alternative is a *universal schedule*, a single connection schedule capable of attaining many Pareto-optimal tradeoff points simultaneously, merely by varying the choice of routing paths. In this work we present the first universal schedules for oblivious routing. Our constructions yield universal schedules which are near-optimal for all possible hop-counts h . The key technical idea is to specialize to a type of connection schedule based on cyclic permutations and to develop a novel Fourier-analytic method for analyzing randomized routing on these connection schedules. We first show that a uniformly random connection schedule suffices with multiplicative error in throughput, and latency optimal up to a $\log N$ factor. We then show that a more carefully designed random connection schedule suffices with additive error in throughput, but improved latency optimal up to only constant factors. This uses a tighter probabilistic bound pertaining to martingales in Banach spaces. Finally, we show that our first randomized construction can be made deterministic using a derandomized version of the Lovett-Meka discrepancy minimization algorithm to obtain the same result.

*This paper was accepted to the ACM-SIAM Symposium on Discrete Algorithms (SODA) 2026. This is a pre-print version.

[†]Cornell University. {sb969,rdk2,slm338,hwr34,rz234}@cornell.edu

[‡]Northeastern University. te.wilson@northeastern.edu

1 Introduction

Reconfigurable networks are a novel communication paradigm in which the pattern of connectivity between hosts varies rapidly over time, creating a virtual topology that determines the paths through which data may be sent as links come in and out of existence. While practitioners are experimenting with these networks in an effort to unlock the energy savings and scalability benefits of replacing traditional packet switches with optical circuit switches [SOA⁺15, BCB⁺20], theoreticians have explored the design space of reconfigurable network architectures to understand the inherent tradeoffs between different design goals [AWS⁺22, WAS⁺23, WAS⁺24]. A key finding of prior theoretical work on reconfigurable networks is the existence of an unbounded number of distinct Pareto-optimal network designs, as the network size tends to infinity. These designs are parameterized by hop-count (i.e., average number of physical links per routing path), permitting progressively lower latency as the hop-count increases. This is because unlike static networks, in reconfigurable networks each edge appears only at specific timesteps. Therefore, latency is defined as the number of timesteps one must wait for each intended physical hop to appear, rather than the number of hops taken. This is formalized in Section 2. As the allowable hop-count increases, this directly increases the number of unique destinations reachable within the same number of timesteps, allowing for lower-latency designs. For workloads with completely homogeneous and predictable latency requirements, this allows network operators to choose the network design that meets their latency requirement using the lowest possible hop-count, which translates into the highest possible throughput.

In reality, datacenter workloads contain a mix of low-latency and high-latency connection requests, which creates additional challenges for the network designer. Existing Pareto-optimal reconfigurable network designs use connection schedules that are fine-tuned to optimize latency for a specific hop-count. To serve heterogeneous workloads, the state of the art [ASW⁺24] is to use time-sharing: connection requests are partitioned into subsets (often called *traffic classes*) with homogeneous latency requirements and, correspondingly, the network’s connection schedule is partitioned into sub-schedules each devoted to serving one of the traffic classes. Time-sharing uses the available bandwidth inefficiently, especially when the fraction of time slots devoted to each sub-schedule isn’t perfectly matched to the fraction of connection requests served by that sub-schedule.

A more flexible and efficient way to do reconfigurable networking would be to use a *universal connection schedule* capable of attaining many (or perhaps all) points on the Pareto frontier of throughput versus latency, merely by varying the choice of routing paths to minimize hop-count within the specified latency bound. The question of existence of universal connection schedules was raised as an open problem in prior theoretical work on reconfigurable networks [AWS⁺22]. In this work, we present constructions of universal connection schedules which are near-optimal in the following respects.

Multiplicative ε -approximation First, we present a randomized construction that, for any network size N , achieves the following guarantee with high probability: for any feasible rate of guaranteed throughput, r , the connection schedule supports a routing scheme that guarantees throughput at least $(1 - \varepsilon)r$, while the maximum latency of the routing paths exceeds the best possible latency by a factor of $O_\varepsilon(\log N)$.

Additive ε -approximation Next, we present a different randomized construction that eliminates the $O(\log N)$ factor in the latency bound by relaxing the throughput guarantee from $(1 - \varepsilon)r$ to $r - \varepsilon$. This follows as a corollary of a more general result showing that for any finite set R of throughput values, there is a universal connection schedule supporting routing schemes for each $r \in R$ that guarantee throughput $(1 - \varepsilon)r$, while the maximum latency of the routing

paths exceeds the best possible latency by a factor of $O_{\varepsilon,R}(1)$, i.e. a constant factor that depends on both ε and R but not on the network size, N .

Deterministic construction Finally, we show that the first aforementioned construction can be made deterministic and computationally efficient.

1.1 Summary of results and techniques

A universal oblivious reconfigurable network (ORN) for a fixed network size N consists of a single connection schedule and a family of oblivious routing protocols for each possible throughput value $r \in [0, \frac{1}{2}]$, each defined by routing paths on that connection schedule, and each of which is near-Pareto optimal. Each of our universal ORN constructions is analyzed by making use of a novel connection between oblivious routing protocols, generating polynomials, and the norm of Fourier coefficient vectors.

We standardize a single oblivious routing protocol that is well-defined on any connection schedule, only requiring two parameters: a hop count h and a phase length Λ . This routing protocol uses exactly $2h$ hops per routing path, and attains maximum latency no more than $(2h+1)\Lambda$. However, it only guarantees throughput $\frac{1}{2h}(1-\varepsilon)$ for some $\varepsilon \geq 0$ which depends on properties of the connection schedule. To show that a single connection schedule is universal, we show that for each possible value of h , this oblivious routing protocol guarantees near-optimal throughput and latency for some carefully chosen phase length Λ dependent on h .

Our oblivious routing protocol is motivated by the packet spraying implementation of Valiant Load Balancing (VLB), with a novel twist. VLB is a technique which routes flow obliviously from source node a to destination node b by routing on shortest paths through a uniformly random intermediate node in the network. VLB is provably optimal in some contexts, notably including reconfigurable networks [AWS⁺22]. In packet spraying, in order to reach a random intermediate node in h hops, you send flow across h *spraying hops* each chosen uniformly at random from a set of Λ potential neighbors. Then you route from that intermediate node to the destination node in h hops, sampled using a time-reversal of the process used to select spraying hops. In a graph with diameter h and suitable expansion properties, this is approximately the same as routing on shortest paths through a uniformly random intermediate node. In our oblivious routing protocol, we packet spray on h -hop paths, with exactly one hop chosen from consecutive non-overlapping phases each of length Λ .

One of our main innovations is departing from traditional VLB to allow the distribution on intermediate nodes to not be uniformly random. We call this framework “VLB with leakage.” We show that if the distribution over intermediate nodes produced by packet spraying is not uniform, but instead only $\frac{\varepsilon}{2}$ -close to uniform, then packet spraying still guarantees throughput $\frac{1}{2h}(1-\varepsilon)$. However, the distribution over intermediate nodes depends on the underlying connection schedule that is used. Another main innovation in our work lies in designing universal connection schedules that enable h -hop packet spraying with an appropriate phase length to generate a nearly uniform distribution for many values of h simultaneously.

As in prior work on reconfigurable network design [AWS⁺22, WAS⁺24, ASW⁺24], we use a type of connection schedule in which the node set is identified with the elements of a finite abelian group, the time steps are identified with a sequence $\dots, s_0, s_1, s_2, \dots$ of group elements, and the node pairs that are connected at time t are those whose difference equals s_t . We call such a structure a *Cayley schedule*. It is analogous to a Cayley graph, except that each edge has an associated time step and paths can only be constructed from edges whose time steps are in increasing order. In prior work designing reconfigurable networks optimized for one specific throughput rate, the Cayley schedules

were based on an h -dimensional vector space over a finite field, where the dimension h was chosen depending on the desired rate of throughput. The generating sequence was then chosen by cycling through the elements of a vector space basis and their scalar multiples. Packet spraying using this design generates the uniform distribution simply because each element of the vector space is uniquely representable as a linear combination of basis elements. However, this design is clearly specialized to one value of h and is not suitable as a universal connection schedule. Instead, in this work we focus on Cayley schedules defined over a *cyclic* group. We refer to these as *shift* connection schedules, since the connectivity pattern in any time step is represented by a cyclic shift permutation.

For shift connection schedules, we examine the generating polynomial of the packet-spraying distribution, a polynomial that is dependent on the shift connection schedule, h , and Λ .

Definition 1. Suppose X is a $\mathbb{Z}/(N)$ valued random variable. Its *generating polynomial* $p_X(z)$ and Fourier coefficient vector $\hat{p}_X[j]$ are given by

$$p_X(z) = \mathbb{E}[z^X] = \sum_{k=0}^{N-1} \Pr(X = k) \cdot z^k, \quad \hat{p}_X[j] = p_X\left(e^{\frac{2\pi i j}{N}}\right). \quad (1)$$

We show that if this Fourier coefficient vector of the packet-spraying distribution (modified by setting the $j = 0$ coordinate to zero) has small 2-norm, this directly implies that packet spraying creates a distribution that is close to uniform. We also show that this Fourier coefficient vector has some nice properties that make its 2-norm easier to bound. It is equal to the element-wise product of averages of Fourier coefficients. We explain this process in further detail in Section 3. We exploit this connection to prove our three main results.

Theorem 1. *Let*

$$L^*(h, N) = hN^{1/h}.$$

Given a constant $\varepsilon > 0$,

1. **(Multiplicative error)** *For sufficiently large N , there exists a randomized algorithm to compute a shift connection schedule σ on N nodes which succeeds with probability at least $\frac{1}{2}$ and is efficiently certifiable. Additionally for each $h \in \{1, \dots, \log_2 N\}$, there also exists an oblivious routing protocol on σ which guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and achieves maximum latency $\tilde{O}(L^*(h, N))$.*
2. **(Additive error)** *For sufficiently large N , there exists a randomized algorithm to compute a shift connection schedule σ on N nodes which succeeds with probability at least $\frac{1}{2}$ and is efficiently certifiable. Additionally for each $h \in \{1, \dots, \log_2 N\}$, there also exists an oblivious routing protocol on σ which guarantees throughput $\frac{1}{2h} - \varepsilon$ and achieves maximum latency $O(L^*(h, N))$.*
3. **(Deterministic, multiplicative error)** *For infinitely many N , there exists a deterministic algorithm to compute a shift connection schedule σ on N nodes which is efficiently computable. Additionally for each $h \in \{1, \dots, \log_2 N\}$, there also exists an oblivious routing protocol on σ which guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and achieves maximum latency $O(L^*(h, N) \cdot \log(N))$.*

Before we summarize the main technical ideas we use to prove each part of Theorem 1, two observations are in order. First, we restrict to hop-counts $h \in \{1, \dots, \log_2 N\}$. This is because for any fixed network size N , one can show that maximum latency cannot be improved beyond $\log_2 N$ using a simple connectivity argument; it is impossible to design a connection schedule that allows

a node a to reach every other node in less than $\log_2 N$ maximum latency. There is an oblivious routing protocol that achieves this maximum latency by sending flow on routing paths that use up to $\log_2 N$ hops. Therefore, increasing hop count beyond $\log_2 N$ can provide no further improvement in maximum latency. Second, there is a lower bound [AWS⁺22] showing $\Theta(L^*(h, N))$ is the optimal latency function for any guaranteed throughput value $\frac{1}{2h}(1 - \varepsilon)$ as long as $\varepsilon \leq 1 - \frac{h}{h+1-(1/C)^h}$ for any constant, C . $\Theta(L^*(h, N))$ is also the optimal latency function for any guaranteed throughput value $\frac{1}{2h} - \varepsilon$ as long as $\varepsilon \leq \frac{1}{2h} \left(1 - \frac{h}{h+1-(1/C)^h}\right)$ for any constant, C .

To prove Theorem 1.1, we consider a shift connection schedule where every shift is chosen independently and uniformly at random. In this case, each individual Fourier coefficient of our generating polynomial can be decomposed into a sum of independently random Fourier coefficients. We apply Hoeffding’s inequality to bound each of these sums with high probability, and then we apply a union bound over Fourier coefficients to derive a bound on the infinity-norm of the Fourier vector, from which a bound on its 2-norm follows. An additional union bound over all possible hop counts h and starting timesteps t in the connection schedule leads to the conclusion that the random shift construction produces a universal schedule with high probability. We find that this analysis requires a phase length Λ_h that is optimal up to $\mathcal{O}_\varepsilon(\log N)$ factors, resulting in an overall latency bound of $\tilde{\mathcal{O}}_\varepsilon(L^*(h, N))$. This analysis is further detailed in Section 4.

For our second randomized result, Theorem 1.2, the main challenge is to remove the $\log(N)$ factor stemming from repeated application of the union bound. Two new ideas are required to accomplish this. First, rather than bounding the infinity-norm of the vector of Fourier coefficients, we bound its q -norm for a value of q that is large but independent of N , using a dimension-independent martingale inequality for the Banach space ℓ^q [Pis16]. This eliminates the union bound over Fourier coefficients. To eliminate the union bound over starting time steps and hop counts, we replace the purely random shift schedule used in Theorem 1.1 with a more complicated construction. We introduce a *convolution product* operation that combines two shift schedules into a longer schedule whose period length is the two product of their two periods. By starting with a purely random base schedule and forming higher and higher powers using the convolution product, we show using Hölder’s Inequality that bounds on the q -norm of the base schedule’s Fourier coefficient vector (for a suitable choice of q) imply bounds on the 2-norm of the Fourier coefficients for the resulting convolved schedule. Bounds on the Fourier 2-norm for every possible starting timestep t , and for any finite number of different hop-counts h , can thereby be derived from a single Fourier q -norm bound for the base schedule, avoiding the use of the union bound. As a corollary, we show this implies additive ε -approximation of throughput for all possible hop counts h , with the correct setting of the base schedule’s period length. This analysis is further detailed in Section 5.

Finally to prove Theorem 1.3, we show a connection between designing a universal shift connection schedule and the high-dimensional vector discrepancy problem. We define a deterministic shift connection schedule by recursively splitting the set of all Fourier coefficient vectors into equal-size sets, maintaining that after each split, the sum of vectors in a set has ∞ -norm only slightly larger than the ∞ -norm of the sum of vectors before the split. This turns out to be possible by relying on a deterministic version of the Lovett-Meka discrepancy minimization algorithm [LRR17], and directly implies a bound on the ∞ -norm of the relevant Fourier coefficient vector for h -hop packet spraying. Then for each value of h , a careful setting of $\Lambda = \mathcal{O}_\varepsilon(N^{1/h} \cdot \log N)$ ensures the resulting distribution is $\frac{\varepsilon}{2}$ -close to uniform. This analysis is further detailed in Section 6.

1.2 Related work

Latency-Throughput Tradeoffs in Reconfigurable Networks. The most important related works [AWS⁺22, WAS⁺23, WAS⁺24] are mentioned above in Section 1. Recent work formally defined oblivious reconfigurable networks (ORNs), and showed the existence of designs that achieve Pareto-optimal tradeoffs between throughput and latency. That is, for any fixed throughput rate r , they constructed a family of infinitely many ORN designs which guaranteed throughput r and achieved optimal maximum latency, up to a constant factor [AWS⁺22]. These results were later extended to any sufficiently large network size [WAS⁺23]. Additionally the Pareto-frontier was significantly improved upon with the introduction of random ORN designs whose guarantees are relaxed to only hold with high probability [WAS⁺24]. Each of these designs uses a connection schedule that is tailored to the tradeoff point they wish to guarantee. Our work aims to construct schedules that capture not just one tradeoff point, but multiple, at the expense of an approximate guarantee.

Valiant Load Balancing (VLB). Valiant and Brebner first introduced oblivious routing in their seminal work [VB81, Val82], demonstrating that their strategy, which became known as Valiant Load Balancing (VLB), achieves optimal latency guarantees over the direct-connect hypercube topology. VLB is the optimal oblivious routing scheme in at least two other contexts: when building a network of fixed-capacity links [KCML05, ZSM05, BC07], and in ORNs [AWS⁺22], in both cases when aiming to fulfill any demand pattern with bounded ingress and egress rates per node. Our work builds upon these results, formalizing a relaxation of VLB that enables guaranteeing approximately optimal throughput.

Practical ORN Proposals. Many proposed data center architectures [GYG⁺17, BCB⁺20, SVB⁺19, AAS23, ASW⁺24] fit the ORN paradigm, some even before it was formally introduced [AWS⁺22]. Rotornet [GYG⁺17] and Sirius [BCB⁺20] propose using optical circuit switches to implement the reconfigurable network fabric, while Shoal [SVB⁺19] proposes using electrical circuit switches. These three designs all attain the same single Pareto-optimal tradeoff point, prioritizing high throughput at the expense of poor latency. Shale [ASW⁺24] uses connection schedules from theoretical work [AWS⁺22], allowing for a tunable tradeoff between throughput and latency, that notably also improved upon resource scaling from previous designs. Mars [AAS23] takes an entirely different approach, motivated by the setting of an ORN with limited buffer memory at each node. Given an arbitrary amount of buffer space, they devise a formula for the ideal emulated degree (or, period) of the schedule, and emulate a de Bruijn graph with this ideal degree. Additionally, more recent work [SAW⁺24] proposes a semi-adaptive reconfigurable connection schedule, relying on the predictability of large-scale structural patterns in datacenter traffic.

Oblivious Routing in General Networks. There is a line of work in oblivious routing primarily focused on minimizing congestion in general networks, compared to an optimal offline routing algorithm. The existence of an oblivious online routing algorithms that obtain a poly-logarithmic competitive ratio was initially proven by Räcke [Räc02]. Later work improved upon this competitive ratio [HHR03], and described polynomial-time constructions that obtain these bounds [ACF⁺03, BKR03, HHR03]. Using multiplicative weights and tree metrics techniques introduced by [FRT04], Räcke further improved the competitive ratio to $O(\log n)$ [Räc08], which was closer to the previously known lower bound of $\Omega(\frac{\log n}{\log \log n})$ [HKLR06]. This oblivious routing algorithm was later combined with centralized traffic engineering solutions to produce a robust and near-optimal system [KYY⁺18]. Although congestion, the focus of much of this line of work, and throughput can be direct inverses of each other, their definition of traffic demands (and therefore throughput) is different from ours, which we take from previous work in reconfigurable network theory [AWS⁺22, WAS⁺23, WAS⁺24]. Additionally, we aim to create an *optimized topology* to do

oblivious routing on, rather than designing algorithms that perform well on an arbitrary topology.

2 Definitions

Definition 2. A *connection schedule* of N nodes and period length T is a sequence of permutations $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{T-1}$, each mapping $[N]$ to $[N]$. $\sigma_k(i) = j$ 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 σ is a directed graph Γ_σ with vertex set $[N] \times \mathbb{Z}$. The edge set of Γ_σ 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 $(i, t) \rightarrow (i, t+1)$ and represent flow waiting at node i during the timestep t . E_{phys} is the set of *physical edges*, which are of the form $(i, t) \rightarrow (\sigma_t(i), t+1)$, and represent flow being transmitted from i to $\sigma_t(i)$ during timestep t . A useful automorphism of Γ_σ is *time-translation*, the function defined on vertices of Γ_σ by $\tau(a, t) = (a, t+T)$. This function maps edges to edges because the connection schedule is T -periodic.

Definition 3. Interpret the set of N nodes to be the commutative group $G = \mathbb{Z}/(N)$ under addition. A *shift connection schedule* on G is a connection schedule where the permutations $\sigma = \sigma_0, \sigma_1, \dots, \sigma_{T-1}$ are all defined by shifts $s_0, \dots, s_{T-1} \in G$, where $\sigma_k(j) = j + s_k$ for all $k \in \{0, \dots, T-1\}$.

We interpret a path in Γ_σ from (a, t) to (b, t') as a potential way to transmit one unit of flow from node a to node b , beginning at timestep t and ending at some timestep $t' > t$. Let $\mathcal{P}$ denote the set of all paths in Γ_σ . Some useful subsets of $\mathcal{P}$ are the following.

$$\begin{aligned} \mathcal{P}_{a,b,t,t'} &= \{\text{paths starting at } (a, t) \text{ and ending at } (b, t')\} \\ \mathcal{P}_{a,b,t} &= \bigcup_{t' > t} \mathcal{P}_{a,b,t,t'} \\ \mathcal{P}_{(a,t) \rightarrow} &= \bigcup_{b \in [N], t > t'} \mathcal{P}_{a,b,t,t'}, \quad \mathcal{P}_{\rightarrow(b,t')} = \bigcup_{a \in [N], t < t'} \mathcal{P}_{a,t,b,t'} \\ \mathcal{P}^e &= \{\text{paths containing edge } e\}. \end{aligned}$$

The time-translation automorphism $\tau : \Gamma_\sigma \rightarrow \Gamma_\sigma$ induces an automorphism of the path set, which we will also denote by τ .

Definition 4. A *flow* is a measure on the measurable space $(\mathcal{P}, 2^\mathcal{P})$. A flow f is called T -periodic if it is invariant under time-translation, i.e. $f(\tau(P)) = f(P)$ for all P . For a flow f and edge e , the amount of flow traversing e is defined to be $f(\mathcal{P}^e)$. We say that f is *feasible* if the amount of flow traversing every *physical* edge is at most 1. Note that in our definition of feasible, we allow virtual edges to have unlimited capacity.

Definition 5. An *oblivious routing protocol* R is a T -periodic flow that defines, for each triple $(a, b, t) \in [N] \times [N] \times \mathbb{Z}$, a unit flow from the starting node (a, t) to the node set $\{(b, t') \mid t' > t\}$. In other words, R satisfies $R(\mathcal{P}_{a,b,t}) = 1$ for all $(a, b, t) \in [N] \times [N] \times \mathbb{Z}$.

Definition 6. A *forward routing distribution* $\mathcal{D}$ is a T -periodic flow that defines, for every starting node and timestep $(a, t) \in [N] \times \mathbb{Z}$, a unit of flow supported on $\mathcal{P}_{(a,t) \rightarrow}$. In other words, $\mathcal{D}(\mathcal{P}_{(a,t) \rightarrow}) = 1$ for all $(a, t) \in [N] \times \mathbb{Z}$. Then $\mathcal{D}_{a,t}(b) = \sum_{P \in \mathcal{P}_{a,b,t}} \mathcal{D}(P)$ denotes the amount of flow assigned to paths which start at (a, t) and end at node b during any ending timestep. Note that for any fixed $(a, t) \in [N] \times \mathbb{Z}$, $\mathcal{D}_{a,t}(b)$ forms a probability distribution over destination nodes b .

A *backwards routing distribution* $\mathcal{E}$ is a T -periodic flow that defines, for every destination node and ending timestep $(b, t') \in [N] \times \mathbb{Z}$, a unit of flow supported on $\mathcal{P}_{\rightarrow(b, t')}$. In other words, $\mathcal{E}(\mathcal{P}_{\rightarrow(b, t')}) = 1$ for all $(b, t') \in [N] \times \mathbb{Z}$. Then $\mathcal{E}_{b, t'}(a) = \sum_t \sum_{P \in \mathcal{P}_{a, b, t, t'}} \mathcal{E}(P)$ denotes the amount of flow assigned to paths which start at node a during any timestep, and end at node b and ending timestep t' . Note that for any fixed $(b, t') \in [n] \times \mathbb{Z}$, $\mathcal{E}_{b, t'}(a)$ forms a probability distribution over starting nodes a .

Definition 7. The *latency* $L(P)$ of a path P in Γ_σ 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 a flow f the *maximum latency* is the maximum over all paths P which may route flow:

$$L_{\max}(f) = \max\{L(P) \mid f(P) > 0\}.$$

Definition 8. A *demand function* D is a function that associates to every source-destination pair $(a, b) \in [N] \times [N]$ and time step $t \in \mathbb{Z}$ a quantity $D(a, b, t) \geq 0$ representing the amount of flow that a aims to send to b starting at time t .

If f is a flow and D is a demand function, the *induced flow* f^D is defined by rescaling f using $D(a, b, t)$ as the scale factor for paths in $\mathcal{P}_{a, b, t}$:

$$f^D(P) = D(a, b, t) \cdot f(P) \quad \forall P \in \mathcal{P}_{a, b, t}.$$

We will typically be applying this rescaling operation when f is a routing protocol or a (forward or backward) routing distribution.

Definition 9. A routing protocol R *guarantees throughput* r if the induced flow R^D is feasible whenever D is a demand function satisfying

$$\forall a \in [N], t \in \mathbb{Z} \quad \sum_{b \in [N]} D(a, b, t) \leq r \quad \text{and} \quad \sum_{b \in [N]} D(b, a, t) \leq r.$$

An equivalent definition [AWS⁺22] requires the induced flow R^D to be feasible for any D that is a set of permutation demands scaled by r . That is, any D for which, for all $t \in \mathbb{Z}$, there exists a permutation $\tau_t : [N] \rightarrow [N]$ such that

$$\begin{aligned} \forall a \in [N], t \in \mathbb{Z}, D(a, \tau_t(a), t) &= r \\ \forall a \in [N], t \in \mathbb{Z}, b \neq \tau_t(a), D(a, b, t) &= 0. \end{aligned}$$

3 Reducing Connection Schedule Design to Fourier Analysis

This section is devoted to proving the following theorem, which relates universal connection schedule design to Fourier analysis.

Theorem 2. *Given a shift schedule $\sigma = s_0, s_1, \dots, s_{T-1}$ on $G = \mathbb{Z}/(N)$, and given parameters Λ and h for which $T \geq \Lambda h$.*

If Λh evenly divides T , then let $\mathcal{A} = \{\ell \Lambda h : \ell \in \mathbb{Z}_{\geq 0}\}$. Otherwise, let $\mathcal{A} = \{0, \dots, T-1\}$. Then for each timestep $t \in \mathcal{A}$, let p_t and q_t be the polynomials

$$p_t(z) = \prod_{j=1}^h p_{t,j}(z), \quad q_t(z) = \prod_{j=h+1}^{2h} q_{t,j}(z)$$

where

$$p_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{s_k} \right), \quad q_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{N-s_k} \right)$$

Define $\hat{p}_t^*$ as the Fourier transform of p with first coordinate set to 0, and $\hat{q}_t^*$ similarly for q .

Now let $\varepsilon > 0$ be a constant. If $\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2}$ and $\|\hat{q}_t^*\|_2 \leq \frac{\varepsilon}{2}$ for all $t \in \mathcal{A}$, then there exists an oblivious routing protocol over σ which guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and maximum latency no more than $2(h+1)\Lambda$.

This section describes our h -spraying-hop-inspired oblivious routing protocol, which we call *Valiant Load Balancing with Leakage*. We first show that this oblivious routing protocol guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ when the forward and backwards spraying hop distributions used to define it are $\frac{\varepsilon}{2}$ -close to uniform. We then devote some time to introducing the Fourier analysis tools used in this paper. Finally, we connect showing that our spraying hop distributions are $\frac{\varepsilon}{2}$ -close to uniform to showing that $\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2}$.

3.1 Valiant Load Balancing with Leakage

Let $\mathcal{D}$ be a forward routing distribution. If we could show that $\mathcal{D}_{a,t}$ forms the uniform distribution over destination nodes for all a, t , then normalizing $\mathcal{D}(P)$ by a factor of N would give an oblivious routing protocol on σ , though not necessarily one with good throughput guarantees. However, if $\mathcal{D}$ was good at routing uniform traffic demands, then we could use the following construction to define an oblivious routing protocol R with certain throughput and latency guarantees.

Definition 10 (Valiant Load Balancing [VB81]). Given an oblivious routing protocol $\mathcal{U}$ with maximum latency L on a N -node connection schedule, we can define a second oblivious routing protocol, $VLB(\mathcal{U})$, motivated by the following informal specification. To route between source-destination pair (a, b) starting at timestep t ,

1. Traffic is routed using $\mathcal{U}$ from (a, t) to a uniformly random intermediate node c in the network.
2. Traffic is routed from the intermediate node $(c, t + L)$ to its final destination b .

We call these two pieces of the routing *semi-paths* of the oblivious routing protocol.

While the informal specification of the routing protocol refers to choosing exactly one intermediate node at random, within the formalism of Definition 4 this would be encoded using a flow which averages over the random choice of intermediate node c . Thus, if $P_0 \in \mathcal{P}_{a,c,t,t+L}$ and $P_1 \in \mathcal{P}_{c,b,t+L,t+2L}$ are two paths and $P = P_0 \circ P_1$ denotes their concatenation, then

$$VLB(\mathcal{U})(P) = \frac{1}{N} \mathcal{U}(P_0) \mathcal{U}(P_1).$$

Theorem 3 (Valiant Load Balancing [VB81]). Let D_{unif} be the uniform traffic demand function with all demands equal to $\frac{1}{N}$. Let $\mathcal{U}$ be an oblivious routing protocol on an N -node connection schedule with maximum latency L such that the induced flow $f^{\frac{1}{N}D_{unif}}$ is a feasible flow. Then $VLB(\mathcal{U})$ is an oblivious routing protocol that achieves maximum latency $2L$ and guarantees throughput $\frac{1}{2h}$.

Requiring $\mathcal{D}$ to form the uniform distribution on destination nodes is quite restrictive. However, we will show that even a distribution that is close to uniform will be good enough to apply something akin to Valiant Load Balancing (VLB).

Definition 11 (h -hop spraying forward and backward routing distributions).

Let σ be a shift connection schedule on $\mathbb{Z}/(N)$ defined by shifts $s_0, s_1, \dots, s_{T-1}$. In addition, suppose T is divisible by h for hop parameter h , and let $\Lambda = T/h$. Partition σ into h consecutive sub-sequences, each of length Λ .

$$\begin{aligned}\sigma_1 &= s_0, \dots, s_{\Lambda-1} \\ \sigma_2 &= s_{\Lambda}, \dots, s_{2\Lambda-1} \\ &\vdots \\ \sigma_h &= s_{(h-1)\Lambda}, \dots, s_{T-1}\end{aligned}$$

We define the forward routing distribution $\mathcal{D}$ given by h -hop packet spraying on phase length Λ . For every path P that first delays flow until the start of a period, then takes exactly one hop from each of the h subsequences $\sigma_1, \dots, \sigma_h$, set $\mathcal{D}(P) = \left(\frac{1}{\Lambda}\right)^h$.

We also define the backwards routing distribution $\mathcal{E}$ formed by h -hop packet spraying in a time-reversed process. Ending at timestep t' , let t^* be the latest timestep that started a period¹ no later than t' . For every path P which starts at timestep $t^* - T$, takes exactly one hop from each of the h subsequences $\sigma_1, \dots, \sigma_h$, and then delays flow at its destination until timestep t , set $\mathcal{E}(P) = \left(\frac{1}{\Lambda}\right)^h$.

Definition 12 (VLB with Leakage). Given a forward routing distribution $\mathcal{D}$ with maximum latency $L_{\mathcal{D}}$ and a backwards routing distribution $\mathcal{E}$ with maximum latency $L_{\mathcal{E}}$, we define the following strategy, which we call *Valiant Load Balancing with Leakage*, which concatenates them into a single oblivious routing protocol with maximum latency $L = L_{\mathcal{D}} + L_{\mathcal{E}}$.

Consider a fixed starting node a , destination node b , and starting timestep t . To define an oblivious routing protocol R , we must define a unit of flow supported on $\mathcal{P}_{a,b,t}$. We will further restrict ourselves to defining a unit of flow supported on $\mathcal{P}_{a,b,t,t'}$ where $t' = t + L_{\mathcal{D}} + L_{\mathcal{E}}$.

Now consider the probability distributions $\mathcal{D}_{a,t}(c)$ and $\mathcal{E}_{b,t'}(c)$. For each intermediate node c , we will first define how to send $\min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t'}(c)\}$ flow from a to b through intermediate node c , on paths concatenated from paths that support $\mathcal{D}$ and $\mathcal{E}$.

To do so, let $t'' = t + L_{\mathcal{D}}$ and find any fractional maximum weighted matching M between $\{P_1 \in \mathcal{P}_{a,c,t,t''} : \mathcal{D}(P_1) > 0\}$ and $\{P_2 \in \mathcal{P}_{c,b,t'',t'} : \mathcal{E}(P_2) > 0\}$, where the edge weights are defined by $w(P_1, P_2) = \min\{\mathcal{D}(P_1), \mathcal{E}(P_2)\}$. Then M has total weight $\min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t'}(c)\}$.² For each pair of paths P_1 in the first set and P_2 in the second set, let $R(P_1 \circ P_2) = M(P_1, P_2)$ be the amount of flow assigned to the concatenation³ of P_1 and P_2 .

In total, we have defined a function which assigned $\sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t'}(c)\}$ flow on routing paths from node a to node b starting at timestep t . Because $\mathcal{D}_{a,t}(c)$ and $\mathcal{E}_{b,t'}(c)$ are probability distributions, this sum is always no more than 1. If it is not equal to 1, then normalize⁴ $R(P)$ for $P \in \mathcal{P}_{a,b,t}$ by this value.

¹In a longer connection schedule, $\mathcal{E}$ paths may be defined over a different set of sub-sequences $\sigma_{\ell}, \dots, \sigma_{\ell+h}$. This is without loss of generality.

²One way to justify this claim is as follows. Subdivide the interval $[0, \mathcal{D}_{a,t}(c)]$ into subintervals $I(P)$ with disjoint interiors, indexed by paths $P \in \mathcal{P}_{a,c,t,t''}$, with $I(P)$ having length $\mathcal{D}(P)$. Similarly, subdivide the interval $[0, \mathcal{E}_{b,t'}(c)]$ into subintervals $J(P)$ with disjoint interiors, indexed by paths $P \in \mathcal{P}_{c,b,t'',t'}$, with $J(P)$ having length $\mathcal{E}(P)$. Then, define $M(P_1, P_2)$ to equal the length of $I(P_1) \cap J(P_2)$.

³Because P_1 ends at $(c, t + L_{\mathcal{D}})$ and P_2 starts at the same place, this concatenated path is well-defined.

⁴The normalization operation is undefined when $\sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t'}(c)\} = 0$, i.e. when the distributions $\mathcal{D}_{a,t}$ and $\mathcal{E}_{b,t'}$ are supported on disjoint sets of nodes. In all of the constructions we propose in this paper, the total variation distance between $\mathcal{D}_{a,t}$ and $\mathcal{E}_{b,t'}$ is strictly less than 1, so they cannot be disjointly supported.

Lemma 1 (VLB with Leakage). *Let $\mathcal{D}$ and $\mathcal{E}$ be the h -hop spraying forward and backwards routing distributions for parameters h and Λ , as defined in Definition 11. Additionally, suppose for every pair of nodes a, b , and pairs of timesteps t, t' , $\mathcal{D}$ and $\mathcal{E}$ form probability distributions over source/destination nodes that are both $\frac{\varepsilon}{2}$ -close to the uniform distribution over N nodes. That is,*

$$\sum_c \left| \mathcal{D}_{a,t}(c) - \frac{1}{N} \right| \leq \frac{\varepsilon}{2}$$

$$\sum_c \left| \mathcal{E}_{b,t'}(c) - \frac{1}{N} \right| \leq \frac{\varepsilon}{2}.$$

Then the routing protocol defined by applying the VLB with Leakage construction to $\mathcal{D}$ and $\mathcal{E}$ achieves maximum latency $2(h+1)\Lambda$ and guarantees throughput $\frac{1}{2h}(1-\varepsilon)$.

Proof. Let $VLB_{\text{leak}}(\mathcal{D}, \mathcal{E})$ be the oblivious routing protocol formed by applying VLB with leakage on $\mathcal{D}$ and $\mathcal{E}$. Since $\mathcal{D}$ and $\mathcal{E}$ both have maximum latency $(h+1)\Lambda$, then clearly $VLB_{\text{leak}}(\mathcal{D}, \mathcal{E})$ will have maximum latency $2(h+1)\Lambda$.

To compute its guaranteed throughput, consider that $\mathcal{D}$ and $\mathcal{E}$ evenly load balance edges in the network and use h hops each per routing path. That is, define the demand function $D(a, c, t) = \frac{1}{h} \mathcal{D}_{a,t}(c)$. Then the induced flow $\mathcal{D}^D$ is a feasible flow. Similarly, define the demand function $E(c, b, t) = \frac{1}{h} \mathcal{E}_{b,t'-2(h+1)\Lambda}(c)$. Then the induced flow $\mathcal{E}^E$ is also a feasible flow.

If we want to route $\mathcal{D}$ and $\mathcal{E}$ at the same time without overloading edge capacities, then we can do so by halving the amount of total flow that each routes. That is, the additive induced flow $\mathcal{D}^{\frac{1}{2}D} + \mathcal{E}^{\frac{1}{2}E}$, where the amount of flow crossing an edge is the sum of both flows crossing the edge, is also a feasible flow.

When we apply the VLB with Leakage construction, we concatenate $\mathcal{D}$ and $\mathcal{E}$ path together, and before normalization, define a function $\overline{VLB}_{\text{leak}}(\mathcal{D}, \mathcal{E})$ that assigns $\sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t'}(c)\} \leq 1$ flow total on routing paths from node a to node b starting at timestep t . $\overline{VLB}_{\text{leak}}(\mathcal{D}, \mathcal{E})$ is not an oblivious routing protocol, because it does not normalize the amount of flow that can be routed between any source-destination pair a, b at timestep t to be 1. However, for any permutation demand function F defined by permutations $\tau_t : [N] \rightarrow [N]$, the induced flow $\overline{VLB}_{\text{leak}}(\mathcal{D}, \mathcal{E})^{\frac{1}{2h}F} \leq \mathcal{D}^{\frac{1}{2}D} + \mathcal{E}^{\frac{1}{2}E}$, which is a feasible flow.

So, what is the minimum amount of flow, η , that $\overline{VLB}_{\text{leak}}(\mathcal{D}, \mathcal{E})$ tells us how to route? Once we lower bound this value, we can use it to derive the guaranteed throughput of the oblivious routing protocol, $VLB_{\text{leak}}(\mathcal{D}, \mathcal{E})$.

To define η more precisely, let

$$\eta = \min_{a,b,t} \left\{ \sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t+4T}(c)\} \right\}.$$

Since $\mathcal{D}$ and $\mathcal{E}$ are T -periodic for a finite T , then there exists an $(a, b, t) \in [N]^2 \times \mathbb{Z}$ for which

$\eta = \sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t+4T}(c)\}$. Fix such an (a, b, t) . Then the following holds.

$$\begin{aligned}
\sum_c \mathcal{D}_{a,t}(c) &= 1 = \sum_c \mathcal{E}_{b,t+4T}(c) \\
\implies 2 &= \sum_c \mathcal{D}_{a,t}(c) + \sum_c \mathcal{E}_{b,t+4T}(c) \\
&= \sum_c \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t+4T}(c)\} + \sum_c \max\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t+4T}(c)\} \\
&= \sum_c (2 \min\{\mathcal{D}_{a,t}(c), \mathcal{E}_{b,t+4T}(c)\} + |\mathcal{D}_{a,t}(c) - \mathcal{E}_{b,t+4T}(c)|) \\
\implies 2 &= 2\eta + \sum_c |\mathcal{D}_{a,t}(c) - \mathcal{E}_{b,t+4T}(c)| = 2\eta + \|\mathcal{D}_{a,t} - \mathcal{E}_{b,t+4T}\|_1.
\end{aligned}$$

Note that we can also bound $\|\mathcal{D}_{a,t} - \mathcal{E}_{b,t+4T}\|_1$ by relying on the fact that $\mathcal{D}$ and $\mathcal{E}$ are $\frac{\varepsilon}{2}$ -close to uniform.

$$\begin{aligned}
\|\mathcal{D}_{a,t} - \mathcal{E}_{b,t+4T}\|_1 &\leq \|\mathcal{D}_{a,t} - \mathbf{u}\|_1 + \|\mathbf{u} - \mathcal{E}_{b,t+4T}\|_1 \\
&= \sum_c \left| \mathcal{D}_{a,t}(c) - \frac{1}{N} \right| + \sum_c \left| \mathcal{E}_{b,t+4T}(c) - \frac{1}{N} \right| \leq \varepsilon,
\end{aligned}$$

where $\mathbf{u}$ is the uniform distribution over N nodes. Plugging everything in, we see that

$$2 \leq 2\eta + \varepsilon \implies \eta \geq 1 - \varepsilon.$$

Recall that $\overline{VL\bar{B}}_{\text{leak}}(\mathcal{D}, \mathcal{E})^{\frac{1}{2h}F}$ is a feasible flow for any permutation demand function F . Then $\frac{1}{\eta} \overline{VL\bar{B}}_{\text{leak}}(\mathcal{D}, \mathcal{E})^{\frac{\eta}{2h}F}$ must also be a feasible flow. And finally, as a flow, $\frac{1}{\eta} \overline{VL\bar{B}}_{\text{leak}}(\mathcal{D}, \mathcal{E}) \geq \overline{VL\bar{B}}_{\text{leak}}(\mathcal{D}, \mathcal{E})$ since η is a lower bound on the values that we normalize by to compute $\overline{VL\bar{B}}_{\text{leak}}$. Therefore, the routing protocol $\overline{VL\bar{B}}_{\text{leak}}(\mathcal{D}, \mathcal{E})$ guarantees throughput $\frac{\eta}{2h} \geq \frac{1}{2h}(1 - \varepsilon)$. $\square$

3.2 Connections to Fourier Analysis

In this subsection we review basic definitions and facts about the discrete Fourier transform. The reader familiar with this material may skip directly to the proof of Theorem 2 at the end of the subsection.

In order to show that our forward and backwards routing distributions are close to uniform, we make use of Fourier analysis on the generating polynomials of our routing distributions. Recall the generating polynomial and Fourier coefficient vector of a distribution on $\mathbb{Z}/(N)$, defined in Equation (1). For independent, integer-valued random variables, the convolution of their generating functions completely characterizes the distribution of their sum. A similar result holds for $\mathbb{Z}/(N)$ valued independent random variables.

Theorem 4. *Suppose X, Y are independent, $\mathbb{Z}/(N)$ valued random variables. Then*

$$p_{X+Y} = p_X p_Y \mod z^N - 1, \quad \text{and} \quad \hat{p}_{X+Y} = \hat{p}_X \odot \hat{p}_Y,$$

where $\mathbf{v} \odot \mathbf{w}$ is the entry-wise product of the vectors $\mathbf{v}$ and $\mathbf{w}$.

Proof. By definition, for $k \in [0, N-1]$,

$$\Pr(X + Y \equiv k \pmod{N}) = \sum_{\substack{i+j \equiv k \\ \pmod{N}}} \Pr(X = i) \Pr(Y = j)$$

Note that as $i, j \in [0, N-1]$, there are two possibilities for each element of the sum on the right: either $i + j = k$ or $i + j = k + N$. However, we have $z^k \equiv z^{k+N} \pmod{z^N - 1}$ which follows from noting that $z^{k+N} - z^k = z^k (z^N - 1)$. Thus,

$$\Pr(X + Y \equiv k \pmod{N}) \cdot z^k \equiv_{(z^N - 1)} \sum_{\substack{i+j \equiv k \\ \pmod{N}}} \Pr(X = i) \Pr(Y = j) \cdot z^{i+j},$$

from which the first desired result follows.

The j th index of the Fourier coefficients of p_{X+Y} is

$$\hat{p}_{X+Y}[j] = p_{X+Y}(\xi^j) = p_X(\xi^j) p_Y(\xi^j) = \hat{p}_X[j] \hat{p}_Y[j]$$

where $\xi = e^{2\pi i/N}$. Therefore, $\hat{p}_{X+Y} = \hat{p}_X \odot \hat{p}_Y$. $\square$

Definition 13. Let $\mathcal{P}(d)$ denote the vector space of polynomials over $\mathbb{C}$ with degree *strictly* less than d .

$$\mathcal{P}(d) = \left\{ \sum_{i=0}^{d-1} p_i z^i \mid p_i \in \mathbb{C}, \forall i \in [d-1] \right\}$$

Suppose $p \in \mathcal{P}(d)$. Then the *Fourier transform* of p is the vector

$$\hat{p} = \begin{pmatrix} p(1) \\ p(e^{2\pi i/d}) \\ p(e^{4\pi i/d}) \\ \vdots \\ p(e^{2\pi i(d-1)/d}) \end{pmatrix}$$

obtained by evaluating p at each of the d -th roots of unity.

Theorem 5 (Parseval's Theorem). Let f be a function over $\mathbb{Z}/(N)$, let $\hat{f}$ is its Fourier transform, an N -dimensional complex vector with entries $\hat{f}_k$. Then

$$\sum_{k \in N} |f(k)|^2 = \frac{1}{N} \sum_{k \in N} |\hat{f}_k|^2.$$

We now re-state and prove the main theorem of this section.

Theorem 2. Given a shift schedule $\sigma = s_0, s_1, \dots, s_{T-1}$ on $G = \mathbb{Z}/(N)$, and given parameters Λ and h for which $T \geq \Lambda h$.

If Λh evenly divides T , then let $\mathcal{A} = \{\ell \Lambda h : \ell \in \mathbb{Z}_{\geq 0}\}$. Otherwise, let $\mathcal{A} = \{0, \dots, T-1\}$. Then for each timestep $t \in \mathcal{A}$, let p_t and q_t be the polynomials⁵

$$p_t(z) = \prod_{j=1}^h p_{t,j}(z), \quad q_t(z) = \prod_{j=h+1}^{2h} q_{t,j}(z)$$

⁵Note that p_t and q_t are generating polynomials of forward and backwards routing distributions which delay flow until timestep t , then use h spraying hops chosen uniformly at random, each from consecutive blocks of Λ timesteps.

where

$$p_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{s_k} \right), \quad q_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{N-s_k} \right)$$

Define $\hat{p}_t^*$ as the Fourier transform of p with first coordinate set to 0, and $\hat{q}_t^*$ similarly for q .

Now let $\varepsilon > 0$ be a constant. If $\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2}$ and $\|\hat{q}_t^*\|_2 \leq \frac{\varepsilon}{2}$ for all $t \in \mathcal{A}$, then there exists an oblivious routing protocol over σ which guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and maximum latency no more than $2(h + 1)\Lambda$.

Proof. Consider our routing distribution $\mathcal{D}$ for connection schedule σ , defined in Definition 11, which first delays flow until a timestep $t \in \mathcal{A}$, then equally weights all paths which take exactly one hop over the next consecutive subsequences each of length Λ . Fixing source node 0, and timestep t , this forms a distribution on destination nodes b with generating polynomial

$$p_t(z) = \prod_{j=1}^h p_{t,j}(z), \quad \text{where} \quad p_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{s_k} \right).$$

This has a Fourier transform that can be decomposed into the element-wise product

$$\hat{p}_t(z) = \bigodot_{j=1}^h p_{t,j}(z),$$

where

$$p_{t,j} = \frac{1}{\Lambda} \begin{pmatrix} 1 \\ \sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} e^{\frac{2\pi i}{N} s_k} \\ \sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} e^{\frac{4\pi i}{N} s_k} \\ \vdots \\ \sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} e^{\frac{(N-1)\pi i}{N} s_k} \end{pmatrix}$$

By assumption, we know that $\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2}$. Since $\mathbf{u}$ is the uniform distribution, $\hat{u} = \mathbf{e}_1$, the first standard basis vector $(1, 0, 0, \dots, 0)^T$. Then $\hat{p}_t^* = \hat{p}_t - \hat{u}$, because the first coordinate of $\hat{p}_t$ equals 1.

$$\hat{p}_t[0] = \bigodot_{j=1}^h \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} e^0 \right) = \bigodot_{j=1}^h \frac{1}{\Lambda} (\Lambda) = 1$$

Therefore,

$$\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2} \implies \|\hat{p}_t - \hat{u}\|_2 \leq \frac{\varepsilon}{2}$$

Use Parseval's Theorem to relate $\|\hat{p}_t - \hat{u}\|_2$ to $\|p_t - u\|_2$, which then bounds $\|p_t - u\|_1$ as desired. For all $t \in \mathcal{A}$,

$$\begin{aligned} \|\hat{p}_t - \hat{u}\|_2 \leq \frac{\varepsilon}{2} &\implies \|p_t - u\|_2 \leq \frac{\varepsilon}{2} N^{-1/2} \\ &\implies \|p_t - u\|_1 \leq \frac{\varepsilon}{2}. \end{aligned}$$

Similarly, consider our backwards routing distribution $\mathcal{E}$ for connection schedule σ , which delays flow backwards until some $t \in \mathcal{A}$ for some non-negative integer ℓ , and then takes one uniformly random hop backwards from each of h consecutive sub-sequences σ_i of length Λ . A similar calculation shows that $\|q_t - u\|_1 \leq \frac{\varepsilon}{2}$ for all $t \in \mathcal{A}$. Apply Lemma 1 to obtain the final result. $\square$

3.3 A Test for Universality

The connections between universal shift connection schedules and Fourier analysis are not unique to the three constructions presented in the following sections of this paper. Given any deterministic shift connection schedule for a fixed network size N , one could numerically test whether such a connection schedule is universal by computing the Fourier coefficient vectors of the generating polynomials directly.⁶ Such a test could even be performed on non-shift connection schedules. However as we discuss in Section 7.2, bounding the closeness to uniformity by computing the Fourier coefficients of a generating polynomial does not work for non-shift connection schedules, leading us to need a different approach.

4 Random Shift Schedule

In this section, we prove that when using h -hop packet spraying and VLB with Leakage, then for a suitable choice of phase length Λ_h , a uniformly random shift connection schedule is a universal connection schedule with constant probability. Applying Theorem 2, we must only prove that for each choice of $h \in \{1, \dots, \log_2 N\}$ and Λ_h , the random Fourier coefficient vector associated with our random shift connection schedule, h , and Λ_h has 2-norm that is bounded by $\frac{\varepsilon}{2}$ with high probability. We will prove this bound for each individual h , and then apply a union bound over all h . That is exactly the focus of the theorem stated below.

Theorem 6. *Given a constant $\varepsilon > 0$, hop parameter h , and a network size N , let $\Lambda \geq 4 \left(\frac{\varepsilon}{2}\right)^{-2/h} N^{1/h} \ln(hN^3)$. Let $\sigma = s_0, \dots, s_{T-1}$ be a sequence of shifts each chosen uniformly at random from $\mathbb{Z}/(N)$, and assume that both $T \leq \frac{N^2}{8 \log_2 N}$ and $T \geq h\Lambda$.*

Then for each timestep $t \in \{0, \dots, T-1\}$, let p_t be the polynomial

$$p_t(z) = \prod_{j=1}^h p_{t,j}(z), \quad \text{where } p_{t,j}(z) = \frac{1}{\Lambda} \left(\sum_{k=t+(j-1)\Lambda}^{t+j\Lambda-1} z^{s_k} \right).$$

Then

$$\Pr \left[\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2} \text{ for every } t \in \{0, \dots, T-1\} \right] \geq 1 - \frac{1}{2 \log_2 N}$$

where $\hat{p}_t^$ is the Fourier coefficient vector of p_t with first entry set to zero.*

Proof. For ease of notation, assume that $\xi = e^{2\pi i/N}$, and note that $p_{t,j}(\xi^a)$ is the a th coordinate of the Fourier coefficient vector of $p_{t,j}$. First suppose for all $j \in [h]$, $t \in [T]$, $a \in [N-1]$, that

$$|\operatorname{Re}(p_{t,j}(\xi^a))| \leq \beta \text{ and } |\operatorname{Im}(p_{t,j}(\xi^a))| \leq \beta$$

for some parameter β . Then it would be the case that

$$\begin{aligned} \|\hat{p}_t^*\|_2^2 &\leq \sum_{a=1}^{N-1} \left(\prod_{j=1}^h p_{t,j}(\xi^a) \right)^2 \\ &\leq N \cdot (\sqrt{2}\beta)^{2h} \end{aligned}$$

⁶Of course, this test can only conclude whether the h -hop packet spraying-inspired VLB with Leakage routing protocols are optimal on this connection schedule for each value h or not. If they are not optimal, that does not imply that no such optimal routing protocols exist for each h . This can be done in polynomial time in the size of the network, N .

If we desire $\|\hat{p}_t^*\|_2 \leq \frac{\varepsilon}{2}$, then setting $\beta = \frac{(\varepsilon/2)^{1/h}}{\sqrt{2}N^{1/2h}}$ suffices. Therefore, in order to bound $\Pr[\|\hat{p}_t^*\|_2 > \frac{\varepsilon}{2}]$, it suffices to bound

$$\Pr\left[|\operatorname{Re}(p_{t,j}(\xi^a))| > \frac{(\varepsilon/2)^{1/h}}{\sqrt{2}N^{1/2h}}\right] \text{ and } \Pr\left[|\operatorname{Im}(p_{t,j}(\xi^a))| > \frac{(\varepsilon/2)^{1/h}}{\sqrt{2}N^{1/2h}}\right],$$

and then union bound over all possible values of t, j , and a , and over the real and imaginary parts.

In order to bound these probabilities, we examine their expected values. By linearity of expectation,

$$\begin{aligned}\mathbb{E}[p_{t,j}(\xi^a)] &= \mathbb{E}\left[\frac{1}{\Lambda} \sum_k \exp\left(\frac{2\pi i}{N} a s_k\right)\right] \\ &= \frac{1}{\Lambda} \sum_k \mathbb{E}_{s_k \in [N]} \left[\exp\left(\frac{2\pi i}{N} a s_k\right)\right] = 0.\end{aligned}$$

Therefore $\mathbb{E}[\operatorname{Re}(p_{t,j}(\xi^a))] = \mathbb{E}[\operatorname{Im}(p_{t,j}(\xi^a))] = 0$ as well.

As each s_k is independently sampled, we can view $\operatorname{Re}(p_{t,j}(\xi^a)) = \sum_k \frac{1}{\Lambda} \operatorname{Re}(\exp(\frac{2\pi i}{N} a s_k))$ and $\operatorname{Im}(p_{t,j}(\xi^a)) = \sum_k \frac{1}{\Lambda} \operatorname{Im}(\exp(\frac{2\pi i}{N} a s_k))$ as sums of Λ random variables, each supported on $[-\frac{1}{\Lambda}, \frac{1}{\Lambda}]$. We then apply Hoeffding's inequality.

Theorem 7 (Hoeffding's Inequality). *Let $X_1, \dots, X_n$ be independent random variables such that $a_j \leq X_j \leq b_j$ almost surely. Then*

$$\Pr\left[\left|\sum_{j=1}^n X_j - \mathbb{E}\left[\sum_{j=1}^n X_j\right]\right| \geq t\right] \leq \exp\left(\frac{-2t^2}{\sum_{j=1}^n (b_j - a_j)^2}\right).$$

Using Hoeffding's inequality, we find that

$$\Pr\left[|\operatorname{Re}(p_{t,j}(\xi^a))| > \frac{(\varepsilon/2)^{1/h}}{\sqrt{2}N^{1/2h}}\right] \leq 2 \exp\left(\frac{-(\varepsilon/2)^{2/h} \Lambda}{4N^{1/h}}\right)$$

and the same tail bound holds for imaginary parts. Union bounding over all possible values of t, j , and a , and over the real and imaginary parts, this shows that

$$\Pr\left[\exists T \in [T], j \in [h] \text{ s.t. } \|\hat{p}_{t,j}^*\|_2 > \frac{\varepsilon}{2}\right] \leq 2TNh \cdot 2 \exp\left(\frac{-(\varepsilon/2)^{2/h} \Lambda}{4N^{1/h}}\right)$$

Assuming $T \leq \frac{N^2}{8 \log_2 N}$ and Λ is the smallest integer for which $\Lambda \geq 4 \left(\frac{\varepsilon}{2}\right)^{-2/h} N^{1/h} \ln(hN^3)$, this implies that

$$\Pr\left[\exists T \in [T], j \in [h] \text{ s.t. } \|\hat{p}_{t,j}^*\|_2 > \frac{\varepsilon}{2}\right] \leq \frac{1}{2 \log_2 N}.$$

□

Corollary 1. *Given a constant $\varepsilon > 0$ and a network size N , let $T = 4 \left(\frac{\varepsilon}{2}\right)^2 N \ln(N^3)$, and let $\sigma = s_0, \dots, s_{T-1}$ be a sequence of shifts chosen uniformly at random from $\mathbb{Z}/(N)$.*

For each $t \in \{0, \dots, T-1\}$ and $h \in [\log_2 N]$, fix $\Lambda_h = 4 \left(\frac{\varepsilon}{2}\right)^{-2/h} N^{1/h} \ln(hN^3)$ and let $p_{h,t}(z)$ be the polynomial

$$p_{h,t}(z) = \prod_{j=1}^h p_{h,t,j}(z), \quad \text{where } p_{h,t,j}(z) = \frac{1}{\Lambda_h} \left(\sum_{k=t+(j-1)\Lambda_h}^{t+j\Lambda_h-1} z^{s_k} \right).$$

Then

$$\Pr \left[\forall h \in [\log_2 N] \text{ and } t \in \{0, \dots, T-1\}, \|\hat{p}_{h,t}^*\|_2 \leq \frac{\varepsilon}{2} \right] > \frac{1}{2}.$$

Proof. For $h \in [\log_2 N]$, define E_h to be the event that $\|\hat{p}_h^*\|_2 > \frac{\varepsilon}{2}$.

By Theorem 6, we know that for $\Lambda_h = 4 \left(\frac{\varepsilon}{2}\right)^{-2/h} N^{1/h} \ln(hN^3)$, we have

$$\Pr[E_h] \leq \frac{1}{2 \log_2 N}$$

Taking the union bound over all $h \in [\log_2 N]$, we get $\Pr[\bigcup_{h=1}^{\log_2 N} E_h] \leq \frac{1}{2}$. □

Corollary 2 (Theorem 1.1). *For sufficiently large N , there exists a randomized algorithm to compute a shift connection schedule σ on N nodes which succeeds with probability at least $\frac{1}{2}$ and is efficiently certifiable. Additionally for each $h \in \{1, \dots, \log_2 N\}$, there also exists an oblivious routing protocol on σ which guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and achieves maximum latency $\tilde{O}_\varepsilon(L^*(h, N))$.*

Proof. The randomized algorithm samples a uniformly random shift schedule $\sigma = s_0, s_1, \dots, s_{T-1}$. The certification procedure works as follows: for every $h \leq \log_2(N)$, compute the vectors $\hat{p}_t^*$ defined in Theorem 2 for every $t < T$, and verify that each of the vectors has 2-norm at most $\frac{\varepsilon}{2}$. By Corollary 1 the random shift schedule passes this test with probability greater than $\frac{1}{2}$, and by Theorem 2, when the test passes the routing protocol defined by VLB with Leakage will satisfy the claimed throughput guarantee. □

5 Convolution Schedule

We devote this section to proving the below theorem. From this, Theorem 1.2 becomes a corollary.

Theorem 8. *Given any set of constants $\{h_1, h_2, \dots, h_\ell\}$ and any constant $\varepsilon > 0$, then for sufficiently large network sizes N there exists a randomized algorithm to compute a shift connection schedule σ on N nodes which succeeds with probability at least $\frac{1}{2}$ and is efficiently certifiable. Additionally for each $j \in \{1, \dots, \ell\}$, there also exists an oblivious routing protocol on σ which guarantees throughput $\frac{1}{2h_j}(1 - \varepsilon)$ and achieves maximum latency $O(L^*(h_j, N))$, where the constant in the $O(\cdot)$ depends only on ε and on the least common multiple of $h_1, \dots, h_\ell$.*

In order to prove this theorem, we describe how to build a shift connection schedule with desirable Fourier properties by convolving smaller shift schedules together, each of which has desirable Fourier properties that will allow us to use Theorem 2. Building up schedules in this way will only guarantee that a constant number of throughput-latency tradeoffs can be achieved on the same connection schedule. However, we will derive a latency bound that is provably optimal up to a constant factor [AWS⁺22], unlike in Section 4. This is because instead of requiring a large union bound, we will be able to make use of the following inequality pertaining to martingales in Banach spaces.

Proposition 1 (Prop 4.35 in [Pis16]). *Let (Ω, A, P) be a probability space. Let Y_n be a sequence of independent mean-zero random variables in Banach space B . Assume that B is of type q , and that the series $\sum_n Y_n$ is a.s. convergent. Then for $0 < \alpha < \infty$, we have*

$$\mathbb{E} \left[\left\| \sum_n Y_n \right\|^\alpha \right] \leq \beta \mathbb{E} \left[\left(\sum_n \|Y_n\|^q \right)^{\alpha/q} \right]$$

where β is a constant that depends only on α and B .

We will now state and prove the following Lemma, which shows the existence of Base Schedules with desirable Fourier properties.

Lemma 2. *Given a constant $\varepsilon > 0$, a positive integer H , let β be a constant that depends on H . Then given a network size N and uniformly random shifts $s_1, \dots, s_\Lambda$ drawn independently from $\mathbb{Z}/(N)$ with $\Lambda \geq \frac{2\beta(2N)^{1/H}}{\varepsilon^{1/H}}$, let polynomial p be*

$$p(z) = \frac{1}{\Lambda} \left(\sum_{k=1}^{\Lambda} z^{s_k} \right)$$

and let $\hat{p}^*$ be the Fourier transform vector of p with first index set to 0. Then

$$\mathbb{E} [\|\hat{p}^*\|_{2H}] \leq \frac{1}{2} \left(\frac{\varepsilon}{2} \right)^{1/H}.$$

Proof. Because the shifts s_k are independent and uniformly distributed, $\hat{p}^*$ is the average of i.i.d. random vectors, each of which is the Fourier transform of a single monomial, z^{s_k} . By padding these N -dimensional vectors with an infinite sequence of 0's, we will consider them as elements $\hat{p}_k^*$ of the infinite-dimensional complex Banach space ℓ^{2H} . The random vectors $\hat{p}_k^*$ have mean $\mathbf{0}$ and are uniformly distributed over the set of N possible Fourier coefficient vectors.

Since $\mathbb{E}[X] \leq \sqrt{\mathbb{E}[X^2]}$ for any random variable X , let us now try to bound $\mathbb{E} [\|\hat{p}^*\|_{2H}^2] = \mathbb{E} \left[\left\| \sum_{k=1}^{\Lambda} \frac{1}{\Lambda} \hat{p}_k^* \right\|_{2H}^2 \right]$. Apply Proposition 1 by interpreting B as ℓ^{2H} , the random variables Y_k as our random vectors $\frac{1}{\Lambda} \hat{p}_k^*$, and $\alpha = 2$. Note that our Banach space has type 2 (and co-type $2H$). We can then derive

$$\mathbb{E} \left[\left\| \sum_{k=1}^{\Lambda} \frac{1}{\Lambda} \hat{p}_k^* \right\|_{2H}^2 \right] \leq \beta \mathbb{E} \left[\sum_{k=1}^{\Lambda} \left\| \frac{1}{\Lambda} \hat{p}_k^* \right\|_{2H}^2 \right]$$

for a constant β that depends only the Banach space ℓ^{2H} and the constant $\alpha = 2$. That is, β is only a function of H .

We may decompose this sum by linearity of expectation.

$$\begin{aligned}
\beta \mathbb{E} \left[\sum_{k=1}^{\Lambda} \left\| \frac{1}{\Lambda} \hat{p}_k^* \right\|_{2H}^2 \right] &= \beta \sum_{k=1}^{\Lambda} \mathbb{E} \left[\left\| \frac{1}{\Lambda} \hat{p}_k^* \right\|_{2H}^2 \right] \\
&= \beta \sum_{k=1}^{\Lambda} \sum_{j=1}^N \frac{1}{N} \left(\sum_{\gamma=1}^{N-1} \left| \frac{1}{\Lambda} \exp \left(\frac{2\pi i}{N} \gamma j \right) \right|^{2H} \right)^{2/2H} \\
&= \beta \sum_{k=1}^{\Lambda} \frac{1}{N} \sum_{j=1}^N \left((N-1) \left(\frac{1}{\Lambda} \right)^{2H} \right)^{1/H} \\
&= \beta \frac{(N-1)^{1/H}}{\Lambda} < \frac{1}{2} \left(\frac{\varepsilon}{2} \right)^{1/H}
\end{aligned}$$

by our assumption that $\Lambda \geq \frac{2\beta(2N)^{1/H}}{\varepsilon^{1/H}}$. □

There is one more theorem to state before we can prove Theorem 8.

Theorem 9 (Hölder's Inequality). *Let $\mathbf{v}_1, \dots, \mathbf{v}_k$ be real or complex-valued vectors, and suppose r and $q_1, \dots, q_k$ are positive numbers such that $\frac{1}{r} = \sum_{j=1}^k \frac{1}{q_j}$. Then*

$$\left\| \bigodot_{j=1}^k \mathbf{v}_j \right\|_r \leq \prod_{j=1}^k \|\mathbf{v}_j\|_{q_j}.$$

We now turn to proving Theorem 8.

Proof. Recall that $\varepsilon > 0$, a network size N , and a set of hop-counts $\{h_1, \dots, h_\ell\}$ are given. Set H equal to the least common multiple of $\{h_1, \dots, h_\ell\}$. Let $\Lambda = \left\lceil \frac{2\beta(2N)^{1/H}}{\varepsilon^{1/H}} \right\rceil$. Then using Lemma 2, we can sample a shift connection schedule with the property

$$\mathbb{E} [\|\hat{p}^*\|_{2H}] \leq \frac{1}{2} \left(\frac{\varepsilon}{2} \right)^{1/H}.$$

Then by Markov's Inequality, with probability at least $\frac{1}{2}$, we sample a schedule with

$$\|\hat{p}^*\|_{2H} \leq 2 \cdot \mathbb{E} [\|\hat{p}^*\|_{2H}] = \left(\frac{\varepsilon}{2} \right)^{1/H}.$$

Call this a *Base Schedule*, and find H Base Schedules⁷ $S_1, \dots, S_H$, each of length Λ , and each with $\|\hat{p}_j^*\|_{2H} \leq \left(\frac{\varepsilon}{2} \right)^{1/H}$. Note that certifying that any given Base Schedule obeys the inequality $\|\hat{p}^*\|_{2H} \leq (\varepsilon/2)^{1/H}$ requires only polynomial time in the number of nodes, N .

Denote Base Schedule $S_j = \{S_j(0), S_j(1), \dots, S_j(\Lambda-1)\}$. Convolve these Base Schedules together to construct a single shift connection schedule with period $T = \Lambda^H$. To do so, represent timesteps as H -tuples, $t = (t_1, \dots, t_H)$, where $t = t_1 + \Lambda t_2 + \dots + \Lambda^{H-1} t_H$. Our shift connection schedule will be defined by the following function: at timestep $t = (t_1, \dots, t_H)$, take shift $s_t = \sum_{j=1}^H S_j(t_j)$.

In addition, for every divisor h of H , we will define an oblivious routing protocol on our shift schedule which achieves maximum latency $\mathcal{O}(hN^{1/h})$ and guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$. We

⁷We may use $S_1 = \dots = S_H$, which only requires sampling a single Base Schedule with the desired properties.

begin with defining the oblivious routing protocol for some special cases of h , and then show the general case.

($h = 1$). In this routing protocol, delay flow until the start of a period. Our forward routing distribution is defined by taking exactly one uniformly random hop during a full period of the schedule. Our backwards routing distribution is defined by taking exactly one uniformly random hop backwards during a full period of the schedule. Use VLB with Leakage to create an oblivious routing protocol with maximum latency no more than $3N = \mathcal{O}(N)$.

To show this routing protocol guarantees throughput $\frac{1}{2}(1 - \varepsilon)$, we will apply Theorem 2. Thus, we need to examine the generating polynomial of the routing distribution, which we denote by $\mathcal{D}$. The generating polynomial p of $\mathcal{D}$ satisfies

$$\begin{aligned} p(z) &= \frac{1}{T} \sum_{t=0}^T z^{st} = \frac{1}{\Lambda^H} \sum_{(t_1, \dots, t_H) \in [\Lambda]^H} z^{\sum_{j=1}^H S_j(t_j)} = \frac{1}{\Lambda^H} \sum_{(t_1, \dots, t_H) \in [\Lambda]^H} \prod_{j=1}^H z^{S_j(t_j)} \\ &= \prod_{j=1}^H \left(\frac{1}{\Lambda} \sum_{\gamma=0}^{\Lambda-1} z^{S_j(\gamma)} \right) = \prod_{j=1}^H p_j(z) \quad \text{where } p_j(z) = \frac{1}{\Lambda} \left(\sum_{\gamma=0}^{\Lambda-1} z^{S_j(\gamma)} \right). \end{aligned}$$

Therefore,

$$\hat{p}^* = \bigodot_{j=1}^H \hat{p}_j^*.$$

In order to apply Theorem 2, we must bound the 2-norm of $\hat{p}^*$. We apply Hölder's Inequality.

$$\begin{aligned} \|\hat{p}^*\|_2 &\leq \prod_{j=1}^H \|\hat{p}_j^*\|_{2H} \\ &\leq \left(\left(\frac{\varepsilon}{2} \right)^{1/H} \right)^H = \frac{\varepsilon}{2} \end{aligned}$$

A similar derivation holds for the backwards routing distribution, and thus, applying Theorem 2 gives our desired guaranteed throughput bound.

($h > 1$). To define the routing protocol we require that H is divisible by h . Let $g = H/h$. In this routing protocol, delay flow until a timestep t with $t_1, \dots, t_g = 0$, and some arbitrary $t_{g+1}, \dots, t_H$. Our forward routing distribution is defined by taking h total hops, exactly one during each of the next h blocks of Λ^g timesteps,

$$[t, t + \Lambda^g - 1], [t + \Lambda^g, t + 2\Lambda^g - 1], \dots, [t + (h-1)\Lambda^g, t + h\Lambda^g - 1].$$

Our backwards routing distribution is defined similarly, by taking exactly one backwards hop during each of $[t + h\Lambda^g, t + (h+1)\Lambda^g - 1]$ through $[t + (2h-1)\Lambda^g, t + 2h\Lambda^g - 1]$. Use VLB with Leakage to create an oblivious routing protocol with maximum latency no more than $\mathcal{O}(h\Lambda^g)$. By our choice of $\Lambda = \left\lceil \frac{2\beta(2N)^{1/H}}{\varepsilon^{1/H}} \right\rceil$, we have $\Lambda^H = \mathcal{O}(N)$ where the (large) constant inside the $\mathcal{O}$ depends on ε and H but not on N . Hence,

$$\mathcal{O}(h\Lambda^g) = \mathcal{O}(h\Lambda^{H/h}) = \mathcal{O}(hN^{1/h}) = \mathcal{O}(L^*(h, N)).$$

To show that this routing protocol guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$, we will apply Theorem 2. According to the proof of that theorem, the generating polynomial $p_t(z)$ of the forward routing

distribution factorizes as $p_t(z) = \prod_{j=1}^h p_{t,j}(z)$ where:

$$p_{t,j}(z) = \frac{1}{\Lambda^g} \sum_{t'=t+(j-1)\Lambda^g}^{t+j\Lambda^g-1} z^{st}.$$

To simplify $p_{t,j}(z)$, recall t is divisible by Λ^g . Write $t + (j-1)\Lambda^g$ in base Λ as

$$t + (j-1)\Lambda^g = \Lambda^g t_{g+1} + \Lambda^{g+1} t_{g+2} + \cdots + \Lambda^{H-1} t_{H-1}. \quad (2)$$

Then

$$\begin{aligned} p_{t,j}(z) &= \frac{1}{\Lambda^g} \sum_{(t_1, \dots, t_g) \in [\Lambda]^g} z^{\sum_{k=1}^H S_k(t_k)} = \frac{1}{\Lambda^g} \sum_{(t_1, \dots, t_g) \in [\Lambda]^g} \prod_{k=1}^H z^{S_k(t_k)} \\ &= \prod_{k=1}^g \left(\frac{1}{\Lambda} \sum_{\gamma=0}^{\Lambda-1} z^{S_k(\gamma)} \right) \cdot \prod_{k=g+1}^H z^{S_k(t_k)} \\ &= \left(\prod_{k=1}^g p_{[k]}(z) \right) \prod_{k=g+1}^H z^{S_k(t_k)} \quad \text{where } p_{[k]}(z) = \frac{1}{\Lambda} \left(\sum_{\gamma=1}^{\Lambda} z^{S_k(\gamma)} \right). \end{aligned} \quad (3)$$

In order to apply Theorem 2, we must bound the 2-norm of $\hat{p}_t^*$. For $j \in \{1, 2, \dots, h\}$ let $f_{t,j}(z) = \prod_{k=g+1}^H z^{S_k(t_k)}$ denote the monomial appearing in the factorization (3), where the digits $t_{g+1}, \dots, t_H$ appearing in the expression for $f_{t,j}(z)$ implicitly depend on t and j via Equation (2). Note that, since $f_{t,j}(z)$ is a monomial in z , every Fourier coefficient $\hat{f}_{t,j}$ is a complex number of norm 1. Hence, for any other vector v ,

$$\|v \odot \hat{f}_{t,j}\|_2 = \|v\|_2.$$

Now, using Hölder's Inequality,

$$\begin{aligned} \|\hat{p}_t^*\|_2 &= \left\| \bigodot_{j=1}^h \hat{p}_{t,j}^* \right\|_2 = \left\| \bigodot_{j=1}^h \left(\hat{f}_{t,j}^* \odot \bigodot_{k=1}^g \hat{p}_{[k]}^* \right) \right\|_2 = \left\| \bigodot_{j=1}^h \bigodot_{k=1}^g \hat{p}_{[k]}^* \right\|_2 \\ &\leq \prod_{j=1}^h \prod_{k=1}^g \|\hat{p}_{[k]}^*\|_{2gh} = \prod_{j=1}^h \prod_{k=1}^g \|\hat{p}_{[k]}^*\|_{2H} \leq \left(\left(\frac{\varepsilon}{2} \right)^{1/H} \right)^{gh} = \frac{\varepsilon}{2}. \end{aligned}$$

A similar derivation holds for the backwards routing distribution, and thus, applying Theorem 2 gives our desired guaranteed throughput bound

In conclusion. Sampling a Base Schedule with the desired Fourier properties occurs with probability at least $\frac{1}{2}$. We convolve our Base Schedule with itself to build a larger shift connection schedule. On this shift connection schedule, for any integer h which divides H , there is an oblivious routing protocol which achieves maximum latency no more than $\mathcal{O}(hN^{1/h}) = \mathcal{O}(L^*(h, N))$ and guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$. □

Corollary 3 (Theorem 1.2). *Given a constant $\varepsilon > 0$, for sufficiently large network sizes N there exists a randomized algorithm to compute a shift connection schedule σ on N nodes which succeeds with probability at least $\frac{1}{2}$ and is efficiently certifiable. Additionally for each $h \in \{1, 2, \dots, \log_2 N\}$ there exists an oblivious routing protocol on that shift connection schedule which guarantees throughput $\frac{1}{2h} - \varepsilon$. and achieves maximum latency $\mathcal{O}(L^*(h, N))$.*

Proof. Note that when $\frac{1}{2h} \leq \varepsilon$, then $\frac{1}{2h} - \varepsilon \leq 0$. Any oblivious routing protocol, even one which does not route any flow, guarantees throughput 0. We therefore only need to handle the case when $\frac{1}{2h} > \varepsilon$.

Let $\{h_1, \dots, h_l\}$ be the set of positive integers with $h < \frac{1}{2\varepsilon}$. As ε is a constant, this set also has constant size. Apply Theorem 8. The routing protocols returned by Theorem 8 guarantee throughput $\frac{1}{2h}(1 - \varepsilon)$. Since $\frac{1}{2h} \leq \frac{1}{2}$, we have $\frac{1}{2h}(1 - \varepsilon) \geq \frac{1}{2h} - \varepsilon$. □

6 Derandomized Construction

In this section, we aim to provide a deterministic, computationally efficient construction of a universal connection schedule. Previous work [AWS⁺22] conjectured that the following deterministic connection schedule inspired by [TBKJ19] would be universal: Let $\mathbb{F}$ denote the finite field with N elements, and let x denote a primitive root in $\mathbb{F}$. Define the sequence of permutations $\pi_0, \pi_1, \dots$ by specifying that $\pi_k(i) = i + x^k$ for all $i \in \mathbb{F}$, $k \in \mathbb{N}$. Notably, the conjectured design is a shift connection schedule, and thus can be tested for individual values of N using Theorem 2 by computing the Fourier coefficient vectors of the generating polynomials directly. However, we make no further progress toward proving this design is universal for infinitely many network sizes N .

Instead, we describe a deterministic construction of a universal shift connection schedule that is motivated directly by derandomizing the techniques we used in Sections 3 and 4. Like in all previous sections, we will build a connection schedule, show that this connection schedule has desirable Fourier properties, and then use Theorem 2 to show the existence of oblivious routing protocols that guarantee any throughput value up to a multiplicative error of $(1 - \varepsilon)$.

At a high level, our strategy is the following recursive approach. We assume N is a power of 2 and construct a universal shift connection schedule with period length $T = N$ in which the shifts $s_0, s_1, \dots, s_{N-1}$ constitute a permutation of the elements of $\mathbb{Z}/(N)$. The permutation is constructed by using a recursive partitioning process to define a total ordering of the elements of $\mathbb{Z}/(N)$. Take the set of all possible shifts to start. Divide this set into 2 equal-size pieces S_{left} and S_{right} , in such a way that the “Fourier 2-norm” does not increase by very much. Our total ordering will place all of S_{left} before all of S_{right} . To order the shifts within each of S_{left} and S_{right} , repeat this process recursively. If we can do the partitioning at each level of the recursion in a deterministic way, then we end up with a deterministic ordering of all possible shifts such that: for any 2^ℓ , if we break $\{0, \dots, N - 1\}$ into $2^{n-\ell}$ consecutive blocks of 2^ℓ shifts, then the Fourier coefficients of each of those blocks of shifts have bounded 2-norm, and thus we can rely on Theorem 2.

In order to do the partitioning deterministically, we rely on discrepancy minimization theory. In particular, a derandomized algorithm that achieves the Spencer’s theorem bound for complex vector discrepancy minimization. This result bounds the ∞ -norm instead of the 2-norm, however we may use that to bound the 2-norm.

6.1 Discrepancy Minimization for Complex Matrices

Levy, Ramdas, and Rothvoss prove the following:

Theorem 10 (Theorem 1 of [LRR17]). *Let $v_1, \dots, v_m \in \mathbb{R}^n$ unit vectors, let $x^{(0)} \in [-1, 1]^n$ be a starting point, and let $\lambda_1 \geq \dots \geq \lambda_m \geq 0$ be parameters so that $\sum_{i=1}^m \exp(-\lambda_i^2/16) \leq n/32$. Then there is a deterministic algorithm that computes a vector $x \in [-1, 1]^n$ with $\langle v_i, x - x^{(0)} \rangle \leq 8\lambda_i$ for all $i \in [m]$ and $|\{i \mid x_i = \pm 1\}| \geq n/2$, in time $\mathcal{O}(\min\{n^4 m, n^3 m \lambda_1^2\})$.*

From this, we get the following derandomization for vector discrepancy too.

Theorem 11. *Let $A \in [-1, 1]^{m \times n}$. Then there exists a universal constant K and a deterministic algorithm that finds an $x \in \{-1, +1\}^n$ such that*

$$\|Ax\|_\infty \leq K \sqrt{n \log \frac{2m}{n}}.$$

Proof. Much of this follows from the proof of Theorem 2 from Lovett-Meka [LM12].

Let $A_k \in \mathbb{R}^n$ denote the k -th row of A . Let $v^k = A_k / \|A_k\|_2$ and define $\alpha(x, y) = \sqrt{80 \ln 2x/y}$. Note that $m \cdot \exp(-\alpha(m, n)^2/16) < n/32$.

The core idea is to run the algorithm from Theorem 10 $\log_2 n$ many times, at each step fixing at least half the coordinates.

Initialize the algorithm from Theorem 10 with $x^0 = \mathbf{0}$. We obtain a vector $x^1 \in [-1, 1]^n$ with $\langle v^k, x^1 \rangle \leq 8\alpha(m, n)$ for all $k \in [m]$ and $|\{j \mid x_j^1 = \pm 1\}| \geq n/2$. Let $S = \{j \mid x_j^1 \neq \pm 1\}$. Then, we run the algorithm again starting at x_S^1 , with the family of unit vectors $\{v^k / \|v^k\|_2 \mid k \in S\}$ and $\lambda_i = \alpha(m, |S|)$ to find a vector x_S^2 with $\langle v_S^k, x_S^2 - x_S^1 \rangle \leq 8 \|v_S^k\|_2 \alpha(m, |S|)$ for all $k \in S$. Repeat this iterative procedure at most $\log_2 n$ times to obtain a vector x so that $|\{j \mid x_j = \pm 1\}| = n$. The discrepancy at the end then is

$$\|Ax\|_\infty \leq \sum_{i=0}^{\log_2 n} \sqrt{n/2^i} \alpha(m, n/2^i) = 8\sqrt{80n} \sum_{i=0}^{\log_2 n} \sqrt{\frac{\ln 2^{i+1}m/n}{2^i}} \leq K \sqrt{n \ln 2m/n}$$

for some universal constant, K , with the last inequality following by noting that the sum can be dominated by a convergent geometric series. $\square$

Corollary 4. *Let $A \in \mathbb{C}^{m \times n}$ with $\|A_{j,k}\|_1 \leq 1$ for all j, k . Then there exists a universal constant K and a deterministic algorithm that finds an $x \in \{-1, +1\}^n$ such that*

$$\|Ax\|_\infty \leq K \sqrt{n \log \frac{4m}{n}}. \quad (4)$$

Furthermore, if n is even, x may be chosen to satisfy $x_1 + \dots + x_n = 0$.

Proof. To create a matrix $D \in \mathbb{R}^{(2m+1) \times n}$, take each entry $a_{j,k}$ of A , split it into its real and imaginary components, and (effectively) replace the entry of A with the 2-dimensional vector $(\text{Re}(a_{j,k}), \text{Im}(a_{j,k}))$, doubling the height of each column of A . Then, augment the matrix with an initial row of 1's, resulting in a matrix D having $2m+1$ rows and n columns. Applying Theorem 11 to D we deduce that there is a vector $\tilde{x} \in \{\pm 1\}^n$ such that

$$\|D\tilde{x}\|_\infty \leq K \sqrt{n \log \frac{4m+2}{n}} \quad (5)$$

If n is odd we set $x = \tilde{x}$. If n is even, the vector $\tilde{x}$ may fail to satisfy the equation $\tilde{x}_1 + \dots + \tilde{x}_n = 0$. Let $\alpha = |\tilde{x}_1 + \dots + \tilde{x}_n|$. Since the first row of D is made up of 1's, the first coordinate of the vector $D\tilde{x}$ is equal to $\pm\alpha$. By Inequality (5) this implies $\alpha \leq K \sqrt{n \log \frac{4m+2}{n}}$. Modify $\tilde{x}$ into a vector x that satisfies $x_1 + \dots + x_n = 0$ by reversing the signs of $\alpha/2$ coordinates. In other words, find a vector $y \in \{-2, 0, 2\}^n$ with exactly $\alpha/2$ non-zero coordinates, such that $x = \tilde{x} + y$ belongs to $\{\pm 1\}^n$ and satisfies $x_1 + \dots + x_n = 0$. We have $\|y\|_1 = 2 \cdot (\alpha/2) = \alpha$, and

$$\|Dx\|_\infty \leq \|D\tilde{x}\|_\infty + \|Dy\|_\infty \leq \|D\tilde{x}\|_\infty + \|y\|_1 \leq 2K \sqrt{n \log \frac{4m+2}{n}}. \quad (6)$$

Inequality (4) above can be justified by redefining the universal constant K in the corollary to be three times the constant K that appears in this proof. $\square$

6.2 Subdividing Columns of the Fourier Matrix

Assume that N is a power of 2, $N = 2^n$. Let $\mathcal{W} \in \mathbb{C}^{N \times N}$ denote the Fourier transform matrix, defined by $\mathcal{W}_{j,k} = e^{\frac{2\pi i}{N}jk}$ for all $j, k \in \{0, \dots, N-1\}$. Let $\mathbf{1} \in \mathbb{C}^N$ denote the vector with all entries equal to one. Given a set $S \subseteq \{0, \dots, N-1\}$, let $\mathcal{W}_S$ denote the submatrix obtained by selecting just the columns in S from $\mathcal{W}$. For ease of notation, when $S = \{j\}$ is a singleton, we just write $\mathcal{W}_j$. We also let $\mathcal{W}^* \in \mathbb{C}^{N \times N}$ denote the Fourier transform matrix with first row entries set to 0.

We will repeatedly apply Corollary 4 to subdivide sets of even cardinality into two subsets of equal size, as depicted in Figure 1. To subdivide a given set S in the diagram, denote the cardinality of S by $2j$ and apply Corollary 4 using the matrix $A = \mathcal{W}_S$, to obtain a vector $x \in \{\pm 1\}^{2j}$ satisfying $x_1 + \dots + x_{2j} = 0$. Then define the two “child sets” of S to be $S^- = \{s \mid x_s = -1\}$ and $S^+ = \{s \mid x_s = +1\}$. The sets produced by this construction obey the following lemma.

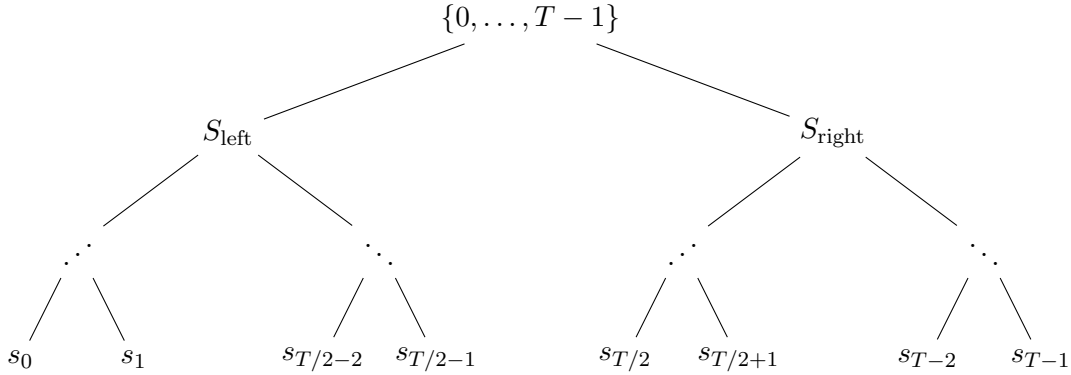


Figure 1: Recursive partitioning scheme for obtaining a total ordering of S

Lemma 3. *Start with the total set $\{0, \dots, N-1\}$, and assume that $N = 2^n$ for some integer n . After using Corollary 4 recursively ℓ times, $\{0, \dots, N-1\}$ is partitioned into 2^ℓ blocks, each containing $2^{n-\ell}$ shifts. Call the total set of these blocks $\mathcal{A}_\ell$. Then for every $A \in \mathcal{A}_\ell$ for $\ell > 0$,*

$$\|\mathcal{W}_A^* \mathbf{1}\|_\infty \leq C \cdot 2^{-\ell/2} \sqrt{N\ell}$$

for a constant C .

Proof. We will show a bound of the form $\|\mathcal{W}_A^* \mathbf{1}\|_\infty \leq M_\ell$ whenever $\ell > 0$ and $A \in \mathcal{A}_\ell$, where the value of M_ℓ is given by a recursive formula. Then we will use the recurrence to show the desired closed formula for M_ℓ .

Claim: When $0 \leq \ell \leq n$, then for every $A \in \mathcal{A}_\ell$, $\|\mathcal{W}_A^* \mathbf{1}\| \leq M_\ell$ for M_ℓ with the recursive formula

$$M_\ell = \begin{cases} 0 & \text{if } \ell = 0 \\ \frac{1}{2}M_{\ell-1} + \frac{1}{2}K \cdot \sqrt{N \cdot 2^{-\ell+1}(\ell+1)} & \text{if } \ell > 0. \end{cases}$$

The claim is proven by induction on ℓ . In the base case, $\mathcal{A}_0$ only contains the set $S = \{0, \dots, N-1\}$, and therefore

$$\|\mathcal{W}_S^* \mathbf{1}\|_\infty = \|\mathcal{W}^* \mathbf{1}\|_\infty = 0.$$

Now consider the recursive case $\ell > 0$. By inductive hypothesis, assume for every $A \in \mathcal{A}_{\ell-1}$, $\|\mathcal{W}_A^* \mathbf{1}\| \leq M_{\ell-1}$. Consider that in our next level, any set $B \in \mathcal{A}_\ell$ is created by first taking some set

$A \in \mathcal{A}_{\ell-1}$ and partitioning it using a vector $x \in \{\pm 1\}^{|A|}$ into equal-sized pieces, A_{left} and A_{right} , one of which equals B . The vector in $\{0, 1\}^{|A|}$ identifying the elements of B is equal to one of the vectors $(\mathbf{1} + x)/2$ or $(\mathbf{1} - x)/2$. By Corollary 4,

$$\begin{aligned} \|\mathcal{W}_B^* \mathbf{1}\|_\infty &\leq \frac{1}{2} (\|\mathcal{W}_A^* \mathbf{1}\|_\infty + \|\mathcal{W}_A^* x\|_\infty) \\ &\leq \frac{1}{2} \left(M_{\ell-1} + K \sqrt{|A| \log_2 (4N/|A|)} \right) \\ &= \frac{1}{2} \left(M_{\ell-1} + K \sqrt{2^{n-\ell+1} \log_2 (4N/2^{n-\ell+1})} \right) \\ &= \frac{1}{2} \left(M_{\ell-1} + K \sqrt{N \cdot 2^{-\ell+1} (\ell+1)} \right). \end{aligned}$$

Now that we have proved the recursive formula of M_ℓ , we use it to find a closed formula.

$$\begin{aligned} M_\ell &= \frac{1}{2} K \sum_{j=1}^{\ell-1} \left(\frac{1}{2} \right)^{\ell-1-j} \sqrt{2^{n-j} (j+1)} \\ &= \frac{1}{2} K \sum_{j=1}^{\ell-1} \sqrt{\frac{2^{n-j} (j+1)}{2^{2(\ell-1-j)}}} \\ &= \frac{1}{2^\ell} K \sum_{j=1}^{\ell-1} \sqrt{2^{n+j} (j+1)} \\ &= \frac{1}{2^\ell} K \sqrt{N} \sum_{j=1}^{\ell-1} \sqrt{2^j (j+1)}. \end{aligned}$$

Finally, we simplify this closed formula to obtain the desired bound.

$$\begin{aligned} M_\ell &\leq K \frac{1}{2^\ell} \sqrt{N(\ell+1)} \sum_{j=1}^{\ell-1} 2^{j/2} \\ &\leq K 2^{-\ell} \sqrt{2N\ell} \cdot \frac{2^{\ell/2} - 1}{\sqrt{2} - 1} \\ &\leq \frac{4K}{\sqrt{2} - 1} \cdot 2^{-\ell/2} \sqrt{N\ell}. \end{aligned}$$

□

6.3 Applying Theorem 2

Theorem 12 (Theorem 1.3). *Let $\varepsilon > 0$ be a constant, and assume $N = 2^n$ is a power of 2. Let $\sigma = s_0, \dots, s_{N-1}$ be the deterministic shift connection schedule created by repeatedly applying Corollary 4 n times on $\{0, \dots, N-1\}$ and the Fourier transform matrix $\mathcal{W}$. Then σ can be efficiently computed. Additionally for any hop count h , there exists an oblivious routing protocol on σ that guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$ and attains maximum latency no more than $O_\varepsilon(L^*(h, N) \cdot \log N)$.*

Proof. Note that for hop count $h = 1$, we may use phase length $\Lambda_1 = N$. In this case, $\hat{p}_{1,t}^* = \mathcal{W}^* \mathbf{1}$, and thus has 2-norm equal to 0.

Let $h > 1$ be some hop count, and let $\Lambda_h = 2^g$ be the smallest power of 2 for which

$$\Lambda_h = 2^g \geq \frac{C^2(4N)^{1/h}}{\varepsilon^{2/h}} \cdot \log_2 N,$$

where C is the universal constant from Corollary 4. Note that Λ_h divides $N = 2^n$. Then applying Theorem 2, we only need to check that $\|\hat{p}_{h,t}^*\|_2 \leq \frac{\varepsilon}{2}$ and $\|\hat{q}_{h,t}^*\|_2 \leq \frac{\varepsilon}{2}$ for all starting timesteps t that are multiples of Λ_h .

Recall that

$$\hat{p}_{h,t}^* = \bigodot_{j=1}^h \hat{p}_{h,t,j}^*$$

where

$$\hat{p}_{h,t,j}^* = \frac{1}{\Lambda_h} \begin{pmatrix} 0 \\ \sum_{k=t+(j-1)\Lambda_h}^{t+j\Lambda_h-1} e^{\frac{2\pi i}{N} s_k} \\ \sum_{k=t+(j-1)\Lambda_h}^{t+j\Lambda_h-1} e^{\frac{4\pi i}{N} s_k} \\ \vdots \\ \sum_{k=t+(j-1)\Lambda_h}^{t+j\Lambda_h-1} e^{\frac{(N-1)\pi i}{N} s_k} \end{pmatrix},$$

and that $\hat{q}_{h,t}^*$ is similarly defined. We will bound their 2-norms by first bounding $\|\hat{p}_{h,t,j}^*\|_\infty$.

Notice that because t is a multiple of $\Lambda_h = 2^g$, then $\hat{p}_{h,t,j}^* = \frac{1}{\Lambda_h} \mathbf{W}_A^*$ for some set $A \in \mathcal{A}_{n-g}$. Then we may use Lemma 3 to bound the ∞ -norm.

$$\begin{aligned} \|\hat{p}_{h,t,j}^*\|_\infty &\leq \frac{1}{\Lambda_h} \cdot C \cdot 2^{-(n-g)/2} \sqrt{N(n-g)} \\ &= C \cdot \left(\frac{1}{\Lambda_h} \cdot 2^{g/2} \right) \cdot \left(2^{-n/2} \cdot \sqrt{N} \right) \cdot \sqrt{n-g} \\ &= C \cdot \frac{1}{\sqrt{\Lambda_h}} \cdot \sqrt{n-g} \\ &\leq \sqrt{\frac{C^2 \log_2 N}{\Lambda_h}} \\ &\leq \frac{(\frac{\varepsilon}{2})^{1/h}}{N^{1/2h}} \end{aligned}$$

where the last line follows from our assumption that $\Lambda_h \geq \frac{C^2(4N)^{1/h}}{\varepsilon^{2/h}} \cdot \log_2 N$. This implies a bound on the $2h$ -norm.

$$\begin{aligned} \|\hat{p}_{h,t,j}^*\|_{2h} &\leq \left(\left(\frac{(\frac{\varepsilon}{2})^{1/h}}{N^{1/2h}} \right)^{2h} \cdot N \right)^{1/2h} \\ &= \left(\frac{\varepsilon}{2} \right)^{1/h}. \end{aligned}$$

We can then apply Hölder's Inequality.

$$\|\hat{p}_{h,t}^*\|_2 = \left\| \bigodot_{j=1}^h \hat{p}_{h,t,j}^* \right\|_2 \leq \prod_{j=1}^h \|\hat{p}_{h,t,j}^*\|_{2h} \leq \left(\left(\frac{\varepsilon}{2} \right)^{1/h} \right)^h = \frac{\varepsilon}{2}.$$

A similar bound can be derived for $\|\hat{q}_{h,t}^*\|_2$. Therefore, applying Theorem 2 gives our desired guaranteed throughput bound $\frac{1}{2h}(1 - \varepsilon)$.

Note that the oblivious routing protocol VLB with Leakage attains a maximum latency of $(2h + 1)\Lambda_h$. Because $\Lambda_h \leq \frac{2C^2(4N)^{1/h}}{\varepsilon^{2/h}} \cdot \log_2 N$, then $(2h + 1)\Lambda_h \leq \mathcal{O}(hN^{1/h} \cdot \log N) = \mathcal{O}(L^*(h, N) \cdot \log N)$. $\square$

7 Conclusion

In this paper, we proved the existence of universal connection schedules for oblivious reconfigurable networks (ORNs). We described three different constructions, two random and one deterministic which each achieved slightly different near-optimal tradeoffs between guaranteed throughput and maximum latency. Below, we discuss some extended results and possible future directions.

7.1 Multiple Traffic Classes, Simultaneously

One motivation of a universal connection schedule is to satisfy heterogeneous workloads containing a mix of low- and high-latency traffic requests. A natural question, then, is how much throughput for each traffic class can be fulfilled simultaneously on the ORN designs we construct in this paper? We limit discussion in this section to the uniformly randomized design of Section 4, though the same ideas apply to our other designs with similar results.

Consider a workload made up of connection requests each with a specified source, destination, starting timestep, and maximum latency. The workload can be partitioned into *traffic classes* indexed by positive integers, with traffic class h representing the subset of connection requests whose maximum latency is between $2(h + 1)\Lambda_{h+1}$ and $2h\Lambda_h$, where $\Lambda_h = \tilde{\mathcal{O}}_\varepsilon(N^{1/h})$ is as defined in Section 4. For workloads whose connection requests all belong to a single traffic class, h , the randomized ORN design of Section 4 guarantees throughput $\frac{1}{2h}(1 - \varepsilon)$. When routing multiple traffic classes simultaneously, let us start by assuming that each traffic class h has its own desired throughput rate r_h , fixed across all time.

Proposition 2. *Consider the randomized ORN design of Section 4, which includes a single connection schedule and oblivious routing protocols R_h for each traffic class h . Let $r_h \in [0, \frac{1}{2h}(1 - \varepsilon)]$ be the desired throughput rate of traffic class h , and suppose that*

$$\sum_h \frac{2h}{1 - \varepsilon} r_h \leq 1.$$

Then each traffic class h may continuously send at rate r_h simultaneously without overloading the network. That is, for every set of demand functions D_h satisfying

$$\forall a \in [N], t \in \mathbb{Z} \quad \sum_{b \in [N]} D(a, b, t) \leq r_h \quad \text{and} \quad \sum_{b \in [N]} D(b, a, t) \leq r_h,$$

the sum of induced flows $\sum_h R_h^{D_h}$ is feasible.

Requiring that each traffic class h maintain the same desired throughput rate across all time is quite restrictive. We may relax that assumption by introducing a desired throughput rate $r_{h,t}$ for traffic class h at time t .

Proposition 3. Let $r_{h,t}$ be the desired throughput rate of traffic class h at timestep t , and suppose that

$$\forall t^* \quad \sum_h \sum_{t \in [t^* - 2h\Lambda_h + 1, t^*]} \frac{r_{h,t}}{(1-\varepsilon)\Lambda_h} \leq 1.$$

Then each traffic class may send at rate $r_{h,t}$ simultaneously without overloading the network. That is, for every set of demand functions D_h satisfying

$$\forall a \in [N], t \in \mathbb{Z} \quad \sum_{b \in [N]} D(a, b, t) \leq r_{h,t} \quad \text{and} \quad \sum_{b \in [N]} D(b, a, t) \leq r_{h,t},$$

the sum of induced flows $\sum_h R_h^{D_h}$ is feasible.

The proofs of Propositions 2 and 3 are simple counting arguments and are thus omitted.

To understand why this more complex relationship exists when traffic classes request varying throughput rates over time, consider the simple case when there are only two traffic classes, class 1 and class 2. Up until timestep t^* , class 1 requests the largest through it can, $\frac{1}{2}(1-\varepsilon)$ while class 2 requests 0. Starting at timestep t^* , class 1 requests 0, while class 2 would like to request the largest throughput it can, $\frac{1}{4}(1-\varepsilon)$. However, class 1 still has traffic in the network that has not yet been fully delivered to its final destination. This means that the full network bandwidth is not immediately available for class 2 to use. It must wait for the class 1 traffic to clear through the network before it can send at the full throughput rate $\frac{1}{4}(1-\varepsilon)$. Instead, at timestep t^* , class 2 may start sending at throughput rate $\frac{(1-\varepsilon)\Lambda_2}{2\Lambda_1}$ for this and all future timesteps. At timestep $t^* + 4\Lambda_2$, the first class 2 traffic begins to reach its destination, and thus class 2 may increase its throughput rate to $\frac{(1-\varepsilon)\Lambda_2}{\Lambda_1}$ for this and all future timesteps. This process of increasing class 2's throughput rate every $4\Lambda_2$ timesteps continues until timestep $t^* + 2\Lambda_1$, when all class 1 traffic has been delivered to its destination and class 2 may finally send at its maximum throughput rate $\frac{1}{4}(1-\varepsilon)$.

7.2 Extending the VLB with Leakage Framework to Other Models

Section 3.1 formalizes a general framework for applying VLB-like routing when flow is not distributed uniformly among all intermediate nodes. We believe this framework is useful in many network settings, not solely for shift connection schedules on ORNs.

The routing distributions from VLB with Leakage take exactly h random hops, with each hop taken uniformly at random from consecutive non-overlapping phases of Λ_h timesteps. At the end of these h hops, we hope to be at an approximately uniformly random node, no matter which source node (or, destination node when considering backwards routing distributions) we started at. Throughout this paper, we limited analysis of VLB with Leakage to shift connection schedules only. Because we dealt with shift connection schedules, we only needed to check that the routing distributions were close to uniformly random for a single source node (or, single destination node). We could then apply the spatial symmetry of shift connection schedules to show this implied close to uniformly random routing distributions for every source (or destination) node. Additionally, one can check this criterion using generating polynomials that are multiplied together. This was the main motivation for limiting to shift connection schedules. The generating polynomial approach does not work when we relax to non-shift connection schedules.

A more general approach that would apply to non-shift connection schedules would be to use Markov transition matrices. For each phase k , one can compute the Markov transition matrix M_k of the phase. Let $M_k[i, j]$ denote the probability that a message starting at node i at the beginning of phase k , and taking a uniformly random hop during the phase, ends at node j . Then M_k is an

average of Λ_h permutation matrices, each representing the permutation of connections for each timestep in the phase. The matrix product $\mathbf{M} = M_h \dots M_2 M_1$ is therefore the probability matrix representing the routing distributions from VLB with Leakage, using h hops with phase length Λ_h . In fact, each row i of $\mathbf{M}$ is exactly the forward routing distribution when starting at source node i . (Or, one could compute a similar matrix product for the backwards routing distributions.) To show that each routing distribution is $\frac{\varepsilon}{2}$ -close to uniform, one must bound the maximum row 1-norm (also known as the ℓ_∞ operator norm) of $\mathbf{M} - U$, where U is the uniform $N \times N$ matrix with all entries equal to $\frac{1}{N}$. Thus, to test whether any fixed non-shift connection schedule is universal when using the routing distributions from VLB with Leakage, one only needs to compute these matrix products for each $h \in \{1, \dots, \log_2(N)\}$ and verify that when U is subtracted, they have low ℓ_∞ operator norm. This can be done in polynomial time in the size of the network.

Proving that a random non-shift connection schedule is universal is more difficult. One would need a way to bound the probability that the product of h matrices, each an average of Λ_h random permutation matrices sampled from a specified distribution, is far from U in ℓ_∞ operator norm. For shift connection schedules, all of the matrices in question are circulant matrices, so one can conjugate by the Fourier transform matrix to simultaneously diagonalize them. This change of basis (which amounts to a reinterpretation of the Fourier-analytic approach we adopted in Section 3) was a key step enabling our analysis of the constructions in this paper. For connection schedules whose constituent permutations belong to some other subgroup Γ of the permutation group S_N , the action of Γ on $\mathbb{R}^N$ splits into a direct sum of irreducible representations, and the analogous change of basis conjugates each permutation matrix represented by Γ to a block-diagonal matrix with one block for each irreducible representation in this decomposition. Can this representation-theoretic perspective, which generalizes our Fourier-analytic approach, lead to useful bounds on the ℓ_∞ operator norms of matrices $\mathbf{M} - U$ for schedules based on non-abelian subgroups $\Gamma \subseteq S_N$? We leave this question to future work.

Static networks

One could even apply the VLB with Leakage framework in the context of static networks. In static networks the hop parameter h translates cleanly, however the analogous parameter to phase length Λ_h is the degree of the static network d . Instead of taking h hops uniformly at random, each from consecutive non-overlapping timesteps, the analogous procedure in a static network would be an h -hop random walk through the network. At the end of this h -hop random walk, in order to use the VLB with Leakage framework, we hope to be at an approximately uniformly random node. Similarly to non-shift connection schedules on ORNs, we can use Markov transition matrices to bound how close we are to uniformly random after an h -hop random walk starting from any node.

In fact, the resulting forward and backwards routing distributions from applying the VLB with Leakage framework to static networks are exactly the Markov chain that represents taking a random walk on a static network. If we take an h -hop random walk in the network, how well mixed (up to what parameter ε) does the resulting distribution end up being? This is the same as the mixing time of the network, although quantifiers have been flipped compared to their usual ordering. From this, we can state the following informal Corollary.

Corollary 5. *Given an unweighted graph G of degree d with $\frac{\varepsilon}{2}$ -mixing time h , there exists an oblivious routing protocol which uses paths of length $2h$ and guarantees maximum congestion no more than $2h \left(\frac{1}{1-\varepsilon} \right)$ so long as traffic demands are “reasonable,” that is nodes do not request to send or receive more demand than their degree, d .*

Mixing times and their connections to oblivious routing protocols have been studied before. In fact, previous work [SS17] noted a similar conclusion, that one can build oblivious routing protocols

with provable guarantees on congestion by concatenating paths carefully sampled from a random walk process. However, they concatenate paths differently, and they express their results as the competitive ratio and in terms of the spectral gap of the graph.

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.
- [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.
- [ASW⁺24] Daniel Amir, Nitika Saran, Tegan Wilson, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. Shale: A practical, scalable oblivious reconfigurable network. In *Proceedings of the ACM SIGCOMM 2024 Conference*, ACM SIGCOMM '24, page 449–464, New York, NY, USA, 2024. Association for Computing Machinery.
- [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.
- [BKR03] Marcin Bienkowski, Mirosław 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.
- [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.
- [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.

- [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.
- [HKLR06] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, Frank Thomson Leighton, and Harald Räcke. New lower bounds for oblivious routing in undirected graphs. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 918–927. ACM Press, 2006.
- [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.
- [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.
- [LM12] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science, FOCS '12*, page 61–67, USA, 2012. IEEE Computer Society.
- [LRR17] Avi Levy, Harishchandra Ramadas, and Thomas Rothvoss. Deterministic discrepancy minimization via the multiplicative weight update method. In Friedrich Eisenbrand and Jochen Koenemann, editors, *Integer Programming and Combinatorial Optimization*, pages 380–391, Cham, 2017. Springer International Publishing.
- [Pis16] Gilles Pisier. *Martingales in Banach Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.
- [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.
- [SAW⁺24] Nitika Saran, Daniel Amir, Tegan Wilson, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. Semi-oblivious reconfigurable datacenter networks. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks, HotNets '24*, page 150–158, New York, NY, USA, 2024. Association for Computing Machinery.
- [SOA⁺15] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, and et al. Jupiter rising: A decade of clos topologies and centralized control in google’s datacenter network. *SIGCOMM Comput. Commun. Rev.*, 45(4):183–197, August 2015.
- [SS17] Michael Schapira and Gal Shahaf. Oblivious routing via random walks, 2017.

- [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.
- [TBKJ19] Edward Tremel, Ken Birman, Robert Kleinberg, and Márk Jelasity. Anonymous, fault-tolerant distributed queries for smart devices. *ACM Trans. Cyber Phys. Syst.*, 3(2):16:1–16:29, 2019.
- [Val82] Leslie G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350–361, 1982.
- [VB81] Leslie G. Valiant and Gordon J. Brebner. Universal schemes for parallel communication. pages 263–277, 1981.
- [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.
- [WAS⁺24] Tegan Wilson, Daniel Amir, Nitika Saran, Robert Kleinberg, Vishal Shrivastav, and Hakim Weatherspoon. Breaking the vlb barrier for oblivious reconfigurable networks. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1865–1876, New York, NY, USA, 2024. 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.