THROUGHPUT AND LATENCY IN RECONFIGURABLE NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Tegan Wilson

August 2024

THROUGHPUT AND LATENCY IN RECONFIGURABLE NETWORKS

Tegan Wilson, Ph.D.

Cornell University 2024

Oblivious routing has a long history in both the theory and practice of networking. In this dissertation, we initialize the formal study of oblivious routing in the context of reconfigurable networks, a new architecture that has recently come to the fore in data center networking, due to its increased energy efficiency and scaling potential. We focus on the tradeoffs between maximizing throughput and minimizing latency in this space.

For every constant throughput rate, we characterize the minimum latency (up to a constant factor) achievable by an oblivious reconfigurable network design. The tradeoff curve turns out to be surprisingly subtle: it has an unexpected scalloped shape, reflecting the fact that routing becomes more costly when average path length is not an integer, since equalizing the path lengths is not achievable. We show that in order to guarantee the throughput value, Valiant load balancing is necessary, which lengthens routing paths by a factor of two. However, we also show that a strictly superior latency-throughput tradeoff is achievable when the throughput bound is relaxed to hold with high probability. The same improved tradeoff is also achievable with guaranteed throughput under time-stationary demands, provided the latency bound is relaxed to hold with high probability and that the network is allowed to be semi-oblivious, using an oblivious (randomized) connection schedule but demand-aware routing.

BIOGRAPHICAL SKETCH

Tegan Wilson grew up in Alexandria, Virginia, just outside the nation's capital. She attended Carleton College, graduating Cum Laude with a double major in Mathematics and Computer Science. She then joined the Computer Science PhD program at Cornell, with a focus on Theoretical Computer Science.

Tegan Wilson is advised by Robert Kleinberg, and is generally interested in algorithms, graph theory, networks and routing, and combinatorics. Her recent work has focused on network and routing designs for reconfigurable datacenter networks, and proving optimal throughput versus latency guarantees in this space.

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Bobby Kleinberg. I arrived to Cornell with a deep love of mathematics, combined with an anxiety that I ultimately would not make it to the end of my PhD. He provided exactly the support I needed, from when I arrived until now. I could not have asked for a better mentor during my time at Cornell.

I would also like to highlight several other research collaborators and faculty mentors who deserve thanks. First and most importantly, Daniel Amir: you were one of my first friends at Cornell. I never thought our research paths would cross as well, and I couldn't be happier to have had that opportunity. I've also enjoyed getting to know Nitika Saran, who has been a tremendous help on our recent projects. I can't wait to see where the PhD takes you.

To my faculty collaborators Hakim Weatherspoon, Vishal Shrivastav, and Rachit Agarwal, thank you for your guidance during the research process. Eva Tardos was the first professor I TA'ed under when coming to Cornell. I learned so much from her and Xanda Schofield, and they both made me feel welcomed almost immediately. Both Eva Tardos and David Williamson, despite their busy schedules, served wonderfully on my special committee. And finally, thank you to the other faculty that supported me along the way, offering advice, attending practice talks, and much more: Eshan Chattopadhyay, Noah Stephens-Davidowitz, Anke van Zuylen, and Michael Kim.

I next want to thank my family. To my parents, Michael and Jennifer: You encouraged my love of mathematics from an early age, and inspired me to attend graduate school. Thank you for all your love and unwavering support over the years. Thank you to my sister Kay, who has always understood me almost better than I've understood myself (for better or worse). Thank you to my grandmother Sue, who not only encouraged me to pursue both mathematics and higher education, but provided an incredible role model, and made it possible for me to reach for those dreams. And thank you to my uncle Ted. I look forward to a future Dr. Wilson party.

Ethan Cecchetti deserves special thanks. Countless times, he provided emotional and

physical support, both during deadlines, and when research and other stressors began to take their toll. The advice he gave me during the late stages of my PhD, and especially during the postdoc job hunt, was incredibly valuable. He's been a blessing to have in my life, first as a wonderful friend, and then a true partner. He has made my life better in so many ways, words would not be able to capture the full extent. Thank you.

Several other Ithaca friends also deserve special thanks. Kristina Wells made me feel welcomed when I arrived in Ithaca. If not for her, I would not have continued swing dancing to nearly the same extent that I do now. And now that she's on the other side of the country, I'm so grateful for our weekly calls. I could always count on Claire Liang to be there for me to rant with when things were going badly. Games and lunches with Eric Campbell and Griffin Berlstein have been a highlight of our friendship. I can't believe Katie Van Koevering and I finally beat Ganon after five years. Thank you for all the delicious cooking you served me during that time. Kate Donahue was always a friendly face, and encouraged me to meet and befriend so much of the community through her large backyard and murder mystery parties. Jasleen Malvai was an amazing friend and roommate, and supported me tremendously during the initial stages of lockdown. Drishti Wali was my first year mentor, and quickly connected me into the CS PhD student community. Ayush Sekhari was both an office-mate and a role model. He comforted me on multiple occasions, and helped me feel like I belonged in Cornell's theory group. Finally, I'm glad to have befriended fellow Bobby Kleinberg students Raunak Kumar and Princewill Okoroafor. (Who else would understand the eccentricities of our advisor?)

To the undergraduate friends that I've kept in touch with: Thank you all for both the support you gave in undergrad, which ultimately led to me arriving at Cornell, and the friendship you continued to give after we scattered across the country. I'd like to thank Jack Wines, Kiya Govek, Valentine Purrell, Vera Vetterli, Zephyr Lucas, and Daniel Lessin in particular for your continued friendship, love, and support.

Some things it's easiest to say among people I've known since I was small. I'm always

happy to see Sarah and CJ Mandell when I return to the DC area. Apart from people I'm related to, you've known me the longest, and still know me unnervingly well. Kristi Thomas, Lesya Melnychenko, and Anjum Choudhury, I've loved continuing to be a part of your lives. Our busy lives have made it difficult, but I'm happy for any excuse to reconnect with you.

And finally, I need to thank Gail, who has supported me so much throughout my PhD. You helped me become the best version of myself. I am so grateful to have met you, and especially grateful that it happened just before lockdown.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

As society moves toward the modern age, data centers are increasingly becoming the backbone of the current technological revolution. Data center demands are expected to triple by 2030 [23], with the expectation that this growth will be met by a combination of increasing capacity within current data centers, and increasing the total number of data centers worldwide.

At a micro level, we can see the effects of this growth playing out in Northern Virginia, currently the largest data center market in the world [14]. Dominion Power, the main electricity provider in the state, cites that data centers contribute to over 20% of their electric sales, almost as much as all other commercial electric sales in the state [19]. In addition, data center capacity in the area is expected to double in size by 2028 [39], which necessitates both new data center buildings, and new electricity infrastructure to support this growth. However, challenges have been mounting due to both the sheer size of the project, and opposition from local jurisdictions and residents about land preservation, environmental, and housing concerns [10, 14].

This motivates studying how to increase data center capacity, without increasing the number of data centers, or their electricity consumption. In this work, we focus on one key component of the data center: the *network*, and we focus on it from a *theoretical* angle.

Data centers are made up of individual computers, or servers, organized into racks, which must *communicate with one another* in order for the data center to function. Thus, it is necessary to build and place physical hardware that enables such communication, and to design communication protocols to effectively use this hardware. Together, these two pieces – the network topology and the routing protocol – constitute a *network design*. Perhaps unsurprisingly, current technology directly informs what network designs are feasible to

implement in practice, how expensive such designs are to build and maintain, and what network designs are considered *optimal* in the space.

When creating an *optimal* network design, the first step is to define what *optimal* means. Common performance metrics in computer networking include *throughput* (equivalently, congestion), or the maximum amount of traffic that may be concurrently sent and delivered within the network, *latency*, or the speed at which traffic can be routed from source to destination, *cost*, which may include either the physical cost of hardware and setup, or upkeep costs such as energy usage, and much more. In addition, there are a host of other properties that we often like our computer networks to have, including (but not limited to): resilience to network node or link failures, memory or space efficiency of the routing protocol, etc. In this thesis, we focus mainly on both *throughput* and *latency*.

Throughput and latency turn out to be fundamentally at odds with each other, which makes them incredibly interesting to study together. Intuitively, this is because optimizing for either throughput or latency requires usage of the same limited resource in the network, total network bandwidth (i.e. total edge capacity). In order for data packets to reach their destination more quickly, more network bandwidth is required per data packet. However, total network bandwidth is a fixed resource. Therefore, routing data more quickly necessitates routing less data in total.

1.1 Reconfigurable Networks

As discussed above, our data center network (and thus our model) is constrained by the real world. For example, it is technologically infeasible to build a network which directly connects every server pair together, because our current networking technology has very limited port counts. Instead, most data centers connect many machines to a piece of technology called a *switch*, typically either a *circuit switch* or a *packet switch*. If machine A wants to send a

message to machine B, and both are connected to the same switch, then machine A simply sends the message to the switch, the message routes through the switch, then to machine B.

Most modern datacenters currently use packet switches, which function in the following way. If machine A wants to send a message directly to machine B, it adds a *header* to the top of the message, which contains the destination information. In this case, the header will indicate that machine B is the intended destination of the message. Then machine A sends the message (including the header) to the packet switch. The packet switch reads the header, and then sends the message to the output port associated with machine B[1].

An alternative is circuit switches, which one can imagine as functioning similarly to old telephone networks; the circuit switch changes (or *reconfigures*) the circuits which directly connect servers to each other. So if machine A wants to send a message directly to machine B, headers are no longer necessary. Instead, machine A needs to know exactly when the circuit switch will set up a circuit connecting it to machine B. This can be done one of two ways. Either the circuit switch has a connection schedule that is predetermined ahead of time, and thus all machines (including machine A) have full knowledge of which machines they will be connected to at which times, and can simply wait their turn. Or, there may be a centralized scheduler. Machine A sends a request to the scheduler that it be connected to machine B, the scheduler computes some schedule based on all the requests it has received thus far, instructs the circuit switch on the new schedule, and additionally informs all machines of the new schedule (including machine A).

Previously, circuit switches had long reconfiguration times. That is, it always took a long time to set up the next circuit in the connection schedule. Therefore, it was desirable to always use a single optimal circuit for the current traffic, and use a multi-hop routing protocol for any traffic that was not directly connected to its destination. This can be seen in

[1]While the specifics of how the packet switch performs this are complex enough to be their own research area, this high-level explanation is all that is necessary for the sake of this document.

early circuit-switched designs for datacenters [20, 35, 50], which relied on predictable traffic demands to choose optimal edge configurations and routes for sending data between nodes. However, this approach impacted latency; this design technique could not handle traffic that required fast routing to its destination, and it provided few other benefits compared to state-of-the-art packet switch networks at the time, so it never went into production.

However, recently circuit switch designs have emerged that are capable of nanosecond-scale reconfiguration times, including both electrical [37] and optical [9, 13, 16] switches. In addition, as network technologies and the desired characteristics of data center deployments continue to evolve, the limitations of packet switches are becoming more apparent. Due to the end of Moore's Law and Dennard Scaling, packet switches face increasing difficulty in scaling to meet network demands without consuming unnecessarily large amounts of power, both within high-density racks [43] and throughout the data center [3]. As a result, many emerging network designs have intentionally avoided using packet switches [15, 17, 21, 24, 26, 34, 36, 40, 44, 49]. Circuit switches present an exciting alternative to packet switches due to their reduced power consumption [3, 43], and potential to scale to arbitrary bandwidth (in the case of optical switches) [9, 36].

Recent works in this space which rely on nanosecond-scale circuit switches [1, 22, 38, 43] have made a case that traffic demands in datacenters are highly unpredictable and change at very fine time granularities, making it challenging, if not impossible, to accurately track demands at any given time. To overcome this fundamental challenge, these works have advocated for network topologies and routing protocols that are *oblivious* to traffic demand matrices. (That is, the circuit switch has a schedule which is predetermined ahead of time, and does not rely on current traffic patterns.)

In this thesis, we focus on a new networking model called *reconfigurable networks*, which theoretically models the networks that are feasible to implement using circuit switch technology. We make the first attempt to formally study the problem of *oblivious routing*, and the inherent

tradeoffs between *throughput* and *latency*, in this context.

1.2 Oblivious Routing

Oblivious routing has a long history in both the theory and practice of networking. By design, an oblivious routing protocol forwards data along a fixed path (or fixed distribution over paths), and is designed to provide good performance guarantees across a wide range of possible traffic demands.

In a landmark 1981 paper, Valiant and Brebner articulated this central problem, and provided a routing strategy that remains state-of-the-art to this day. This solution, which came to be known as *Valiant load balancing*, or *VLB*, was beautifully simple: to send data from source s to destination t, sample an intermediate node u uniformly at random. Then form a routing path from s to t by concatenating "direct paths" from s to u and from u to t. (The definition of direct paths may depend on the network topology; often shortest paths suffice.) This lengthens routing paths by a factor of two and thus consumes twice as much bandwidth as direct-path routing. However, crucially, it is *oblivious*: the distribution over routing paths from s to t depends only on the network topology, not the communication pattern [47].

The focus of oblivious routing research spurred by Valiant and Brebner in the 1980's was on network topologies designed to enable efficient communication among a set of processors, such as hypercubes and shuffle exchange networks [12, 28, 45–47]. These topologies tended to be highly symmetric (often with vertex- or edge-transitive automorphism groups) and tended to have low diameter and no sparse cuts. One could loosely refer to this class of networks as *optimized topologies*.

A second phase of oblivious routing research, initiated by Räcke in the early 2000's,

focused on oblivious routing schemes for *general topologies*. Compared to optimized topologies, the oblivious routing schemes for general topologies required much greater overprovisioning, inflating the capacity of each edge by at least a logarithmic factor compared to the capacity that would be needed if routing could be done using an optimal (non-oblivious) multicommodity flow. This line of work [5, 6, 11, 25, 41] culminated in Räcke's discovery of oblivious routing schemes for general networks that are guaranteed to approximate the optimum congestion within a logarithmic factor in the worst case [42]. This algorithm, which uses a path budget parameter k, was later implemented and tested in wide-area networks for small k, and shown to have good performance [32].

In addition to fully oblivious routing, partially adaptive (or, semi-oblivious) routing protocols have also been examined, in which the router precommits to a limited set of paths between each pair of vertices, and at runtime may only send flow on one of the precommitted paths. When precommiting to only $\log(n)$ paths, this approach was implemented and shown to be effective in wide-area networks [32], and was even recently proven to be polylog(n)-competitive [52]. Since oblivious routing under the same sparsity constraint provably cannot be polylog(n)-competitive [28], to the best of our knowledge, this constitutes the first asymptotic separation between the power of semi-oblivious and oblivious routing.

1.3 Content Overview

In this thesis, we revisit the study of oblivious routing for a new class of networks, *reconfigurable networks*, and investigate the inherent tradeoffs between throughput and latency in this model. Specifically, we ask

> *For every throughput rate r, what is the minimum latency achievable by a reconfigurable network design that achieves throughput r?*

We start with investigating fully oblivious reconfigurable network (ORN) designs which guarantee their throughput value, and we fully resolve this question to within a constant factor[2] for d-regular reconfigurable networks, except when d is very large — bounded below by a constant power of N, the number of nodes in the network. Our optimal network designs use VLB in the construction of their routing protocol, demonstrating that, like networks of fixed-capacity links permitting any communication pattern with bounded ingress and egress rates per node [7, 29, 51], VLB is also a provably optimal technique in reconfigurable networks.

In Chapter 3, we discuss how to build ORN designs which optimally trade off between throughput and latency. Specifically, for each fixed value of throughput r, we show how to build ORN designs for infinitely many network sizes N which guarantee throughput r, and achieve optimal maximum latency (up to a constant factor). This chapter is split into two parts, one for each family of ORN designs which together create this result. The Elementary Basis Scheme (EBS) is optimal for most values of throughput, and the Vandermonde Basis Scheme (VBS) is optimal for the rest.

The designs in Chapter 3, while theoretically optimal, only work for very limited network sizes, which depend on the desired throughput guarantee. In Chapter 4, we show how to extend our ORN designs to any sufficiently large network size N, using a scheme based around "dummy nodes." Like Chapter 3, this chapter is also split into two parts. One for the EBS extended design, and one for the VBS extended design.

In Chapter 5, we present all theoretical lower bounds on latency within this document.

[2] One could, of course, ask the transposed question: *for every latency bound L, what is the maximum guaranteed throughput rate achievable by an oblivious routing scheme with maximum latency L?* Our work also resolves this question, not only to within a constant factor, but up to an additive error that tends to zero as $N \to \infty$. Optimizing throughput to within a factor of two, subject to a latency bound, is much easier than optimizing latency to within a constant factor subject to a throughput bound. The importance of the latter optimization problem, *i.e.* our main question, is justified by the high cost of overprovisioning networks, which leads data center network operators to be much less tolerant of suboptimal throughput than of suboptimal latency.

These lower bounds are necessary to ensure that the reconfigurable network designs described in the rest of the document are optimal. In Section 5.1, we show that the ORN designs from Chapters 3 and 4 are optimal up to a constant factor.

We then show that the ability to *randomize the network topology* in reconfigurable networks allows oblivious routing schemes that improve upon VLB. We obtain reconfigurable network designs that improve upon the maximum latency achievable for a given throughput value by nearly the square root, under two relaxations of obliviousness:

1. In Chapter 6, when the network is allowed a small probability of violating the throughput guarantee; or
2. In Chapter 7, when the throughput guarantee must hold with probability 1, but routing is only *semi-oblivious*.

The proof of the result in Chapter 6 requires a complicated tail bound described in Section 6.4, and we also prove that the topology produced forms an expander graph in Section 6.5.

As noted above in Section 1.2, semi-oblivious routing refers to routing protocols in which the network designer must pre-commit (in a demand-oblivious manner) to a limited set of routing paths between every source and destination, but the decision of how to distribute flow over those paths is made with awareness of the requested communication pattern. In the context of reconfigurable networks, we interpret this to mean that the connection schedule is oblivious but that the routing protocol may be demand-aware. In fact, the semi-oblivious routing protocol that we refer to in Chapter 7 is demand-aware in a very limited sense: it uses the oblivious routing protocol from Chapter 6 with high probability, but in the unlikely event that this leads to congestion on one or more edges, it reverts to using a different oblivious routing scheme that is guaranteed to avoid congestion at the cost of incurring higher latency.

In Section 5.2, we show that the designs from Chapters 6 and 7 are optimal up to a

logarithmic factor in latency. By combining the results from Section 5.3 and Section 7.4, we also prove that purely oblivious reconfigurable network designs (even with a randomized connection schedule) cannot achieve the same result as our semi-oblivious design: if the throughput guarantee must hold with probability 1, then the *average* latency must be strictly asymptotically greater for oblivious reconfigurable networks than for semi-oblivious ones. Thus, we prove an asymptotic separation between the power of semi-oblivious and oblivious routing in reconfigurable networks.

In Chapter 2, we present the formal definitions relevant to the technical chapters of this thesis. We define Oblivious Reconfigurable Network (ORN) designs, Semi-Oblivious Reconfigurable Network (SORN) designs, throughput, and latency. We also discuss and motivate some of the theoretical assumptions we make about reconfigurable network designs throughout the rest of this document.

CHAPTER 2

DEFINITIONS

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

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

The *emulated graph* of the connection schedule $\boldsymbol{\pi}$ can be viewed as a *time-compressed* version of $\boldsymbol{\pi}$. It is a directed graph G_{em} with vertex set $[N]$. The edge set of G_{em} is the set of all edges that appear at some point during the period of $\boldsymbol{\pi}$, that is,

$$E(G_{em}) = \big\{(i, \pi_t(i)) : t \in \{0, \ldots, T-1\}\big\}.$$

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

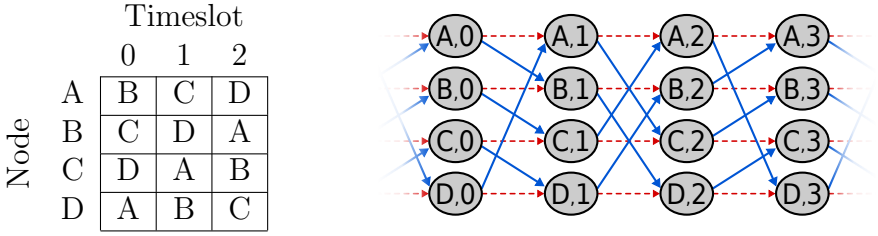

Figure 2.1: A connection schedule among four nodes, as well as part of its corresponding virtual topology. The full virtual topology represents a countably infinite number of timeslots.

$\boldsymbol{\mathcal{P}} = \bigcup_{a,b,t} \boldsymbol{\mathcal{P}}(a,b,t)$ denote the set of all paths in $G_{\boldsymbol{\pi}}$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Definition 10. An ORN Design $\mathcal{R}$ *guarantees throughput* r if the induced flow $f(R, rD)$ is feasible whenever for all t, the row and column sums of $D(t)$ are bounded above by 1. (Such matrices $D(t)$ are called *doubly sub-stochastic*.) An ORN Design $\mathcal{R}$ guarantees throughput r *with respect to time-stationary demands* if for every time-stationary demand function D with row and column sums bounded by 1, then the induced flow $f(R, rD)$ is feasible. An easy application of the Birkhoff-von Neumann Theorem establishes the following: in order for an ORN design to guarantee throughput r with respect to time-stationary demands, it is necessary and sufficient that it guarantee throughput r with respect to permutation demands.

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

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

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

Definition 12. A distribution over SORN designs $\mathscr{S}$, is said to *achieve maximum latency* L

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

Definition 13. A *round robin* for a group of nodes $S = \{s_0, \ldots, s_{k-1}\}$ of size k starting at timestep t_0, is a schedule of $k-1$ timesteps in which each element of S has a chance to send directly to each other element exactly once. Specifically, $\pi_t(s_i) = s_{i+t \pmod k}$ for $t_0 < t < t_0 + k - 2$. That is, during timestep t node s_i may send to node $s_{i+t \mod k}$.

2.1 Assumptions

Note that the definitions in this section are based on the following implicit assumptions.

Fractional flow and no queueing. We interpret the amount of flow traversing an edge as an expected number of packets. We assume that when sending flow from a source to a destination, was may divide that flow into arbitrarily small fractional quantities which may be sent on multiple routes.

Due to this assumption, the ORN and SORN designs described in this document send small fractions of flow from multiple paths across the same link. However in a real system, only one packet from one path may traverse the link during a single timestep. As a result, in real systems queuing may happen, which is best addressed using a congestion control system. Congestion control has a decades-long history of active research across various networking contexts. My collaborators and I discuss this in [4], but I leave discussion of that work to my collaborator Daniel Amir and his dissertation.

No propagation delay. We assume that the total quantity of flow scheduled to be transmitted over a link in one timeslot is received by the end of that timeslot.

In addition, our ORN and SORN models could be enhanced to take propagation delay into account by adjusting the virtual topology. Rather than connecting physical edges from (i, s) to $(j, s + 1)$, they could instead connect to $(j, s + d_{ij})$, where d_{ij} is a whole number representing the propagation delay from i to j in units of timeslots. As in our basic model, nodes of the virtual topology in this enhanced model would be constrained to belong to at most one incoming and at most one outgoing physical edge, though if d_{ij} varies with i and j then the set of physical edges would no longer be described by a sequence of permutations.

2.2 Allowing degree $d > 1$ in a timeslot

Although our formalization of ORNs only describes networks in which nodes have a degree of 1 in every timeslot, it can be generalized to networks that support a d-regular connectivity pattern in each timeslot. When $d > 1$, we interpret a demand matrix D which requests throughput r as one in which the row and column sums of D are bounded above by dr.

To generalize our model of ORNs to allow degree $d > 1$ in a timeslot, one would once again model the virtual topology as a graph with vertex set $[N] \times \mathbb{Z}$ whose edges are partitioned into virtual and physical edges. As before, virtual edges connect (i, t) to $(i, t + 1)$ for all i and t. Physical edges form a T-periodic sequence of d-regular bipartite graphs on vertex set $[N] \times \{t, t + 1\}$ as t varies over $\mathbb{Z}$. The connectivity of $[N] \times \{t, t + 1\}$ is d-regular bipartite. By Kőnig's Theorem, this edge set can be decomposed into d edge-disjoint perfect matchings, which we use to "unroll" into d consecutive timeslots of a 1-regular ORN. Therefore, a d-regular ORN design which guarantees throughput r with maximum latency L unrolls into a 1-regular ORN design which guarantees throughput r with maximum latency dL. Under this

framework, a lower bound $L^*_{orn}(r, N)$ for 1-regular ORN designs trivially implies the lower bound $\frac{1}{d}L^*_{orn}(r, N)$ for d-regular designs.

However, an upper bound for 1-regular designs does not necessarily imply a similar upper bound for d-regular designs, because the routing scheme could route paths containing two or more physical edges in timeslots belonging to the same "unrolled" segment of the 1-regular virtual topology. This would correspond to traversing two or more edges at once in the d-regular topology. Our upper bound constructions found in Chapters 3, 6 and 7 can be easily modified to avoid this problem. Specifically, they can be modified to never allow flow to be routed along two edges within any block of d consecutive time slots, provided $d \leq N^{1/c}$ for a sufficiently large constant c. This modification adds a factor of at most 2 to the maximum latency.

Each design divides the connection schedule into *phases*[1]. In a single routing path, only one physical hop may be taken per phase. If d evenly divides the size of the phases, then the design needs no modification. Otherwise, we may double the length of the connection schedule by iterating through each phase twice in a row. Either routing paths always use the first copy of each phase, or the second copy. This modification clearly both doubles the length of routing paths, and ensures that routing paths never use two edges within any block of d consecutive time slots, provided d is no more than the length of the phases, which can be bounded by $N^{1/c}$ for a sufficiently large constant c.

Then, by inverting the unrolling process, we obtain a d-regular ORN design with maximum latency $L = O\left(\frac{1}{d}L^*_{orn}(r, N)\right)$. This confirms that the tight bound on maximum latency for d-regular ORN designs is $\Theta\left(\frac{1}{d}L^*_{orn}(r, N)\right)$ whenever $d \leq N^{1/(h+1)}$ and justifies our focus on the case $d = 1$ throughout the remainder of this paper.

[1]See Chapters 3, 6 and 7 for more details.

CHAPTER 3

ORN DESIGNS

This chapter is devoted to proving the following theorem.

Theorem 1. *Consider any constant $r \in (0, \frac{1}{2}]$. Let (h, ε) to be the unique solution in $\mathbb{N} \times (0, 1]$ to the equation $\frac{1}{2r} = h + 1 - \varepsilon$, and let $L^*_{orn}(r, N)$ be the function*

$$L^*_{orn}(r, N) = h\left(N^{1/(h+1)} + (\varepsilon N)^{1/h}\right).$$

1. *Then for every $N > 1$ and every ORN design on N nodes that guarantees throughput r, the maximum latency is at least $\Omega(L^*_{orn}(r, N))$.*
2. *Furthermore, for infinitely many network sizes N there exists an ORN design on N nodes that guarantees throughput r and whose maximum latency is $O(L^*_{orn}(r, N))$.*

As a lower bound, Theorem 1.1 is restated and proved in Section 5.1. Our proof of Theorem 1.2 is split into two families of ORN designs, each of which we describe formally in Sections 3.1 and 3.2. However, we first provide a high-level sketch of the main technical ideas behind our designs.

Our design is easiest to describe when the throughput $r = \frac{1}{2h}$ and $N = p^h$ for positive integer h and prime number p. In that case, we use a design that we call the *Elementary Basis Scheme* (EBS) which identifies the set of N nodes with elements of the group[1] $(\mathbb{Z}/(p))^h$. Let $\mathbf{e}$ be the elementary basis consisting of the columns of the $h \times h$ identity matrix. EBS uses a connection schedule whose timeslots cycle through the nonzero scalar multiples of the elementary basis, hence the name *Elementary Basis Scheme*. In a timeslot devoted to $s \cdot \mathbf{e}_i$, the network is configured to allow each node a to send to $a + s \cdot \mathbf{e}_i$. Over the course of one

[1]This should be thought of as the h-dimensional vector space over $\mathbb{Z}/(p)$. While we describe taking the elementary basis for simplicity here, the EBS scheme itself does not require $\mathbb{Z}/(p)$ to be a field, thus we use the word group here. This is further described in Section 3.1.1.

complete cycle, any two nodes can be connected by a "direct path" consisting of h physical hops (or fewer) that modify the coordinates of the source node one by one until they match the coordinates of the destination. The EBS routing protocol constructs a random path connecting a given source and destination using VLB: it chooses a random intermediate node and concatenates two "semi-paths": the direct paths from the source to the intermediate node and from the intermediate node to the destination.

To generalize this design to all non-integer values of $\frac{1}{2r}$, we need to enhance EBS so that a constant fraction of semi-paths use h physical hops and a constant fraction use $h+1$ physical hops. This necessitates a modified ORN design that we call the *Vandermonde Basis Scheme* (VBS). Assume $r = h + 1 - \varepsilon$ for $h \in \mathbb{N}, 0 < \varepsilon < 1$, and that $N = p^{h+1}$ for prime p, so that the nodes can be identified with the vector space $\mathbb{F}_p^{h+1}$. Instead of one basis corresponding to the identity matrix, we now use a sequence of distinct bases each corresponding to a different Vandermonde matrix. In addition to the single-basis semi-paths (which now constitute $h+1$ physical hops), this enables the creation of "hop-efficient" semi-paths composed of h physical hops belonging to two or more of the Vandermonde matrices in the sequence. Hop-efficient semi-paths have higher latency than direct paths, but we opportunistically use only the ones with lowest latency to connect a subset of terminal pairs, joining the remaining pairs with direct semi-paths. A full routing path is then defined to be the concatenation of two random semi-paths, as before.

Proving that the VBS routing protocol guarantees throughput r boils down to quantifying, for each physical edge e, the net effect of shifting load from direct paths that use e to hop-efficient paths that avoid e and vice-versa. The relevant sets of paths in this calculation can be parameterized by unions of affine subspaces of $\mathbb{F}_p^{h+1}$, and the use of Vandermonde matrices in the connection schedule gives us control over the dimensions of intersections of these subspaces, and thus over the size of their union.

We discuss the Elementary Basis Scheme (EBS) in Section 3.1, and the Vandermonde

Basis Scheme (VBS) in Section 3.2. When combined, EBS and VBS give a tight upper bound on maximum latency for all constant guaranteed throughput values r, and prove Theorem 1.2. We address how EBS and VBS can be modified for d-regular networks with $d > 1$ in Section 3.3.

3.1 Elementary Basis Scheme

3.1.1 Connection Schedule

In EBS's connection schedule, each node participates in a series of sub-schedules called round robins. Consider a cyclic group $H = \mathbb{Z}/(p)$ acting freely on a set S of n nodes, where we denote the action of $t \in H$ on $i \in S$ by $i + t$. A round robin for S is a schedule of $p - 1$ timeslots in which each element of S has a chance to send directly to each other element exactly once; during timeslot $t \in [p-1]$ node i may send to $i+t$. The number of round-robins in which each EBS node participates is controlled by a tuning parameter h which we refer to as the *order*. Similar to the previous section, h will be half of the maximum number of physical hops in an EBS path.

Let $p = N^{1/h}$, so that the node set $[N]$ is in one-to-one correspondence with the elements of the group H^h. Each node $a \in [N]$ is assigned a unique set of h coordinates $(a_0, a_1, ..., a_{h-1}) \in H^h$ and participates in h round robins, each containing the n nodes that match in all but one of the h coordinates. We refer to these round robins as *phases* of the EBS schedule. One full iteration of the EBS schedule, or *epoch*, contains h phases. Because each phase is a round robin among n nodes, each phase takes $p - 1$ timeslots, resulting in an overall epoch length of $T = h(p - 1) = h(N^{1/h} - 1)$.

We now describe the EBS schedule formally. We express each node $\boldsymbol{i}$ as the h-tuple

Node	Timeslot 0	1	2	3
A,A	B,A	C,A	A,B	A,C
B,A	C,A	A,A	B,B	B,C
C,A	A,A	B,A	C,B	C,C
·	·	·	·	·
·	·	·	·	·
B,C	C,C	A,C	B,A	B,B
C,C	A,C	B,C	C,A	C,B

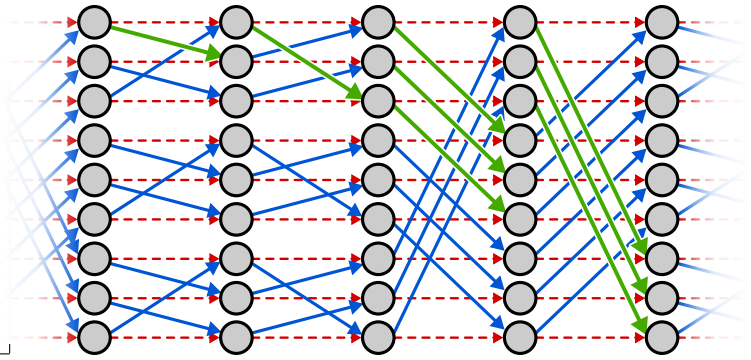

Figure 3.1: Connection schedule for 9 nodes in $h = 2$ EBS, as well as part of the corresponding virtual topology. Physical edges used on semi-paths from ((A,A),0) to other nodes are highlighted in green.

$(i_0, i_1, \ldots, i_{h-1}) \in (\mathbb{Z}/p)^h$. Similarly, we identify each permutation π_k of the connection schedule using a scale factor s, $1 \leq s < p$, and a phase number x, $0 \leq x < h$, such that $k = (p-1)x + s - 1$. Let $\mathbf{e}_x$ denote the standard basis vector whose x^{th} coordinate is 1 and all other coordinates are 0. The connection schedule is then $\pi_{(p-1)x+s-1}(\boldsymbol{i}) = \boldsymbol{i} + s\mathbf{e}_x = \boldsymbol{j}$. Since $\mathbf{e}$ is the standard basis, $j_y = i_y$ for $y \neq x$, and $j_x = i_x + s \pmod{p}$.

The EBS schedule can be seen as simulating a flattened butterfly graph between nodes [31]. This schedule generalizes existing ORN designs which have thus far all been based on the same schedule: a single round robin among all nodes, simulating an all-to-all graph. When $h = 1$, the EBS schedule reduces to this existing schedule. On the other hand, when $h = \log_2(N)$, the EBS schedule simulates a direct-connect hypercube topology. By varying h, in addition to achieving these two known points, the EBS family includes schedules which achieve intermediate throughput and latency tradeoff points.

3.1.2 Oblivious Routing Scheme

The EBS oblivious routing scheme is based around Valiant load balancing (VLB) [47]. VLB operates in two stages: first, traffic is routed from the source to a random intermediate node in the network. Then, traffic is routed from the intermediate node to its final destination.

This two-stage design ensures that traffic is uniformly distributed throughout the network regardless of demand. We refer to the path taken during an individual stage as a *semi-path*, and we use the same algorithm to generate semi-paths in either stage.

To create a semi-path between a node (a, t) and (b, t') for some $t' \approx t + T$, the following greedy algorithm is used starting at (a, t): for the current node in the virtual topology, if the outgoing physical edge leads to a node with a decreased Hamming distance to b (i.e. it matches b in the modified coordinate), traverse the physical edge. Otherwise, traverse the virtual edge. This algorithm terminates when it reaches a node (b, t') for some t'. Note that because there are h coordinates, the largest Hamming distance possible is h, and the longest semi-paths use h physical links, and take no more than T timesteps to complete.

In order to construct a full path from (a, t) to (b, t^*) for some $t^* \approx t + 2T$, first select an intermediate node c in the system uniformly at random. Then, traverse the semi-path from (a, t) to (c, t'), where t' is the timeslot at which the semi-path reaches node c. If $t' < t + T$, traverse virtual edges until node $(c, t + T)$ is reached. Finally, traverse the semi-path from $(c, t + T)$ to (b, t^*).

The EBS oblivious routing scheme is formed as follows: for $R_{a,b,t}$, for all intermediate nodes c, construct the path from (a, t) to (b, t^*) via c as described above, and assign it the value $\frac{1}{N}$. Assign all other paths the value 0. Because there are N possible intermediate nodes, each of which is used to define one path from (a, t) to (b, t^*), this routing scheme defines one unit of flow.

3.1.3 Latency-Throughput Tradeoff of EBS

Proposition 1. *For each $r \leq \frac{1}{2}$ such that $h = \frac{1}{2r}$ is an integer, and each $N > 1$ such that $N^{1/h}$ is an integer, the EBS design of order h on N nodes guarantees throughput r and has*

maximum latency $\frac{1}{r}\left(N^{2r}-1\right)$.

The proof of Proposition 1 is contained in the following two subsections, which address the latency and throughput guarantees respectively.

Latency

Recall that $h = \frac{1}{2r}$ and that $p = N^{1/h} = N^{2r}$, so the latency bound in Proposition 1 can be written as $2h(p-1)$. Since the epoch length is $T = h(p-1)$, the latency bound asserts that every EBS routing path completes within a time interval no greater than the length of two epochs. An EBS path is composed of two semi-paths, so we only need to show that each semi-path completes within the length of a single epoch.

Let (a,t) denote the first node of the semi-path. If t occurs at the start of a phase, then after x phases have completed the Hamming distance to the semi-path's destination address must be less than or equal to $t-x$; consequently the semi-path completes after at most h phases, as claimed. If t occurs in the middle of a phase using basis vector $\mathbf{e}_x$, let s denote the number of timeslots that have already elapsed in that phase. Either the semi-path is able to match the p^{th} destination coordinate before the phase ends, or the coordinate can be matched during the first s timeslots of the next phase that uses basis vector $\mathbf{e}_p$. In either case, the p^{th} destination coordinate will be matched no later than timeslot $t+T$, and all other destination coordinates will be matched during the intervening phases.

Throughput

Lemma 1. *Given an arbitrary demand function D requesting throughput $r = \frac{1}{2h}$ on N nodes, we may generate demand function $\hat{D}'$ with the following properties:*

1. *for all* $t \in [N]$, $\hat{D}'(t)$ *has row and column sums exactly equal to* r

2. $\hat{D}'(t)$ *bounds* $D(t)$ *above.*

Proof. We can generate $\hat{D}'$ by greedily increasing matrix entries by the maximum amount possible, while still maintaining the property that row and column sums are no more than r, until all row and column sums exactly equal r. Due to the latter condition, it follows that $f(R, \hat{D}')$ bounds $f(R, D)$ above; thus $F(f(R, \hat{D}'), e) \geq F(f(R, D), e)$. Henceforward, we focus on proving $F(f(R, \hat{D}'), e) \leq 1$. □

Lemma 2. *Let* R *be the EBS routing scheme for a given* N *and* h*. For all demand functions* D *requesting throughput at most* $\frac{1}{2h}$*, the flow* $f(R, D)$ *is feasible.*

Proof. Due to Lemma 1, we solely focus on the case when arbitrary demand function D has row and column sums exactly equal to $r = \frac{1}{2h}$. Consider an arbitrary physical edge $e \in E_{\text{phys}}$ from (i, t_e) to $(j, t_e + 1)$, where t_e is the timeslot during which the edge begins. Let $t_e \equiv (x_e, s_e)$ such that x_e is the phase in the schedule corresponding to t_e, and s_e is the scale factor used during t_e. We wish to show that $F(f(R, D), e) \leq 1$.

Valid paths in EBS include two components: the semi-path from the source node to an intermediate node, and the semi-path from the intermediate node to the destination node. We can therefore decompose the paths in $F(f(R, D), e)$ into two components as follows: first, we define $\hat{R}'$, a routing protocol defined such that $\hat{R}'_{a,b,t}(P)$ equals 1 if P is the semi-path from (a, t) to (c, t') for some t', and 0 otherwise. Because EBS uses the same routing strategy for both source-intermediate semi-paths and intermediate-destination semi-paths, $\hat{R}'$ is used for both components. Then, we introduce two demand functions: $\hat{D}'_{a\to c}$ represents demand on semi-paths from origin nodes to intermediate nodes, while $\hat{D}'_{c\to b}$ represents demand on semi-paths from intermediate nodes to destination nodes. Note that for all physical edges e,

$$F(f(R, D), e) = F(f(\hat{R}', \hat{D}'_{a\to c}), e) + F(f(\hat{R}', \hat{D}'_{c\to b}), e).$$

To characterize $\hat{D}'_{a\to c}$, note that regardless of source and destination, R samples intermediate nodes uniformly. Therefore, for all $(t, a, c) \in \mathbb{Z} \times [N] \times [N]$,

$$\hat{D}'_{a\to c}(t, a, c) = \frac{1}{N} \sum_{u\in[N]} D(t, a, u) = \frac{r}{N}$$

Similarly, because semi-paths from an intermediate node to the destination always commence exactly T timeslots after the starting vertex, we can characterize $\hat{D}'_{c\to b}(t, b, c)$ as follows:

$$\hat{D}'_{c\to b}(t, c, b) = \frac{1}{N} \sum_{u\in[N]} D(t - T, u, b) = \frac{r}{N}$$

Note that $\hat{D}'_{a\to c} = \hat{D}'_{c\to b} = \hat{D}^{ALL}$, where $\hat{D}^{ALL}$ is the uniform all-to-all demand function $\hat{D}^{ALL}(t, a, b) = \frac{r}{N}$ for all $(t, a, b) \in \mathbb{Z} \times [N] \times [N]$. Therefore, $F(f(R, D), e) \leq 2F(f(\hat{R}', \hat{D}^{ALL}), e)$.

Claim 1. *For all $e \in E_{phys}$, there are exactly Tp^{h-1} triples (t, a, c) such that the semi-path from (a, t) to (c, t') (for some t') traverses e.*

Proof of claim. Denote the endpoints of edge e by (i, t_e) and $(i + s \cdot \mathbf{e}_x, t_e + 1)$. The semi-path of a triple (t, a, c) traverses e if and only if the semi-path first routes from (a, t) to (i, t_e), and $(c - a)_x = s$.

Because semi-paths complete in T timeslots, only semi-paths beginning in timeslots in the range $[t_e - T + 1, \ldots, t_e]$ could possibly reach node (i, t_e) and traverse e. For every $t \in [t_e - T + 1, \ldots, t_e]$, where $t \equiv (x_t, s_t)$, we can construct p^{h-1} such triples as follows: First, we select $\boldsymbol{d}$, a vector representing the difference between a and c in the triple we will construct. To satisfy the second condition on (t, a, c), we must set $\boldsymbol{d}_x = s$. However, the remaining $h - 1$ indices of $\boldsymbol{d}$ can take on any of the n possible values. Thus, there are p^{h-1} possibilities for $\boldsymbol{d}$.

For any semi-path (t, a, c) such that $c - a = \boldsymbol{d}$, the timeslots in which a physical edge is traversed can be determined from $\boldsymbol{d}$. For any given timeslot $t' \equiv (x', s')$ such that $t \leq t' < t+T$, a physical edge is traversed if and only if $\boldsymbol{d}_{x'} = s'$. These are the edges that decrease the

Hamming distance to b by correctly setting coordinate p. We thus construct a as follows: For every index x, if $(\boldsymbol{d}_x, x)$ is between k_t and $k_e - 1$ inclusive, we set $a_p = \boldsymbol{i}_x - \boldsymbol{d}_x$. Otherwise, we set $a_x = \boldsymbol{i}_x$. Once we have constructed a, c is simply $a + \boldsymbol{d}$. This choice of a and c ensures that by timeslot t_e, the semi-path from (a, t) to (b, t') reaches network node i by the phase before timeslot t_e.

For each of the T timeslots for which semi-paths originating in the given timeslot may traverse e, there are p^{h-1} such semi-paths. This gives a total of Tp^{h-1} semi-paths that traverse e over all timeslots. Note that because each such semi-path has a unique $(t, \boldsymbol{d})$, none of the constructed semi-paths are double counted. In addition, because the $(t, \boldsymbol{d})$ pair determines the timeslots in which physical links are followed, and because there is only one physical link entering and leaving each node during each timeslot, there cannot be more than one choice of a for a given $(t, \boldsymbol{d})$ pair such that the semi-path includes (i, t_e). Because the Tp^{h-1} count includes all possible choices of $\boldsymbol{d}$ for every timeslot, all semi-paths that traverse e are accounted for. □

Now we continue with the proof of Lemma 2. Since exactly Tp^{h-1} triples (t, a, c) correspond to semi-paths that traverse e, and $\hat{D}^{ALL}$ assigns $\frac{r}{N}$ flow to each semi-path, $F(f(\hat{R}', \hat{D}^{ALL}), e) = \frac{r}{N} Tp^{h-1} = \frac{r}{N} h(p-1)p^{h-1}$. Thus:

$$\begin{aligned} F(f(R, D), e) &\leq 2F(f(\hat{R}', \hat{D}^{ALL}), e) = 2\frac{r}{N} h(p-1)p^{h-1} \\ &< 2\frac{r}{N} hp^h = 2\frac{r}{N} h(N^{1/h})^h = 2rh \end{aligned}$$

When $r \leq \frac{1}{2h}$, for all physical edges e, $F(f(R, D, e)) \leq 1$. Thus, $f(R, D)$ is feasible. □

3.1.4 Tightness Guarantees

Lemma 3. *For $0 < r \leq \frac{1}{2}$ let $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. The EBS design of order h attains maximum latency at most $C \cdot L^*_{orn}(r, N)$, except when*

$$\varepsilon \geq 2\sqrt{\frac{2h}{\pi}} \left(\frac{2e}{C}\right)^h.$$

Proof. Theorem 1.1 (proved in Section 5.1) and Proposition 1 together show the following about the maximum latency of EBS compared to the maximum latency lower bound:

$$L_{EBS} \leq 2hN^{1/h}$$

$$L^*_{orn}(r, N) \geq \frac{h}{e}(\varepsilon N)^{1/h} \left(\frac{\sqrt{\frac{\pi h}{2}}}{4h}\right)^{1/h}$$

Note that this interpretation of the maximum latency lower bound is taken from equation (5.1) in the proof of Theorem 1.1.

Suppose we wish to assert $L_{EBS}/L^*_{orn}(r, N) \leq C$. Given C and h, we will derive the possible values of ε for which this assertion holds.

$$C \geq \frac{2hN^{1/h}}{\frac{h}{e}(\varepsilon N)^{1/h}\left(\frac{\sqrt{\frac{\pi h}{2}}}{4h}\right)^{1/h}} = \frac{2e}{\left(\frac{\varepsilon\sqrt{\pi h/2}}{4h}\right)^{1/h}}$$

$$\frac{\varepsilon\sqrt{\pi h/2}}{4h} \geq \left(\frac{2e}{C}\right)^h$$

$$\varepsilon \geq 2\sqrt{\frac{2h}{\pi}} \left(\frac{2e}{C}\right)^h.$$

□

When ε falls outside this range, the maximum latency of the EBS design is far from optimal. In the following sections we present and analyze an ORN design which gives a tighter upper bound when ε is very small and falls outside this range, in other words when $\varepsilon < 2\sqrt{\frac{2h}{\pi}} \left(\frac{2e}{C}\right)^h$.

3.2 Vandermonde Basis Scheme

In order to provide a tight bound when ε is very small, we define a new family of ORN designs which we term the Vandermonde Basis Scheme (VBS). VBS is defined for values of N which are perfect powers of prime numbers. We begin by providing some intuition behind the design of VBS.

For $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$, a small value of ε indicates that r is slightly above $\frac{1}{2(h+1)}$. This indicates that the average number of physical hops in a path can be at most slightly below the even integer $2(h+1)$. EBS is only able to achieve an average number of physical hops equal to an even integer as N becomes sufficiently large. In small ε regions, the difference between the highest average number of physical hops theoretically capable of guaranteeing r throughput and the average number of physical hops used by EBS approaches 2. This suggests that EBS achieves a throughput-latency tradeoff that favors throughput more than is necessary in these regions, penalizing latency too much to form a tight bound. A more effective ORN design for these regions would use paths with $2(h+1)$ physical hops, but mix in sufficiently many paths with fewer physical hops to ensure that the average number of physical hops per path is at most $2(h+1-\varepsilon)$.

VBS achieves this by employing two routing strategies for semi-paths alongside each other. The first strategy, single-basis (SB) paths, resembles the semi-path routing used by EBS for $h' = h + 1$. The second strategy, hop-efficient (HE) paths, will rely on the fact that VBS's schedule regularly modifies the basis used to determine which nodes are connected to one another. HE paths will consider edges beyond the current basis, enabling them to form semi-paths between nodes using only h hops, even when this is not possible within a single basis. The more future phases are considered, the more nodes can be connected by HE paths. This tuning provides a high granularity in the achieved tradeoff between throughput and latency, and enables a tight bound in regions where ε is small.

We define VBS for $N = p^{h+1}$ such that p is a prime number. The connection schedule and routing algorithm of VBS depend on a parameter δ, which represents a target for the fraction of semi-paths that traverse HE paths. We later describe how to set Q, the number of future phases considered for HE path formation, such that the number of destinations reachable by HE paths is approximately δN.

3.2.1 Connection Schedule

Before describing the connection schedule of VBS, it is instructive to revisit the schedule of EBS. EBS's schedule consists of h' phases. Each of these phases is defined based on an elementary basis vector $\mathbf{e}_x$, connecting each node $\boldsymbol{i}$ to nodes $\boldsymbol{i} + s\mathbf{e}_x$ for all possible nonzero scale factors s. VBS is defined similarly, except instead of elementary basis vectors, Vandermonde vectors (to be defined in the next paragraph of this section) are used to form the phases. In addition, rather than using a single basis, the VBS connection schedule is formed from a longer sequence of phases, with any set of $h+1$ adjacent phases corresponding to a basis.

As in EBS, each node a is assigned a unique set of $h+1$ coordinates $(a_0, a_1, ..., a_h)$, each ranging from 0 to $p-1$. This maps each node to a unique element of $\mathbb{F}_p^{h+1}$. We identify each permutation π_k of the connection schedule using a scale factor s, $1 \leq s < p$ and a phase number x, $0 \leq x < p$, such that $k = (p-1)x + s - 1$. Each phase p is formed using the Vandermonde vector $\boldsymbol{v}(x) = (1, x, x^2, ..., x^h)$. This produces the connection schedule $\pi_{(p-1)x+s-1}(\boldsymbol{i}) = \boldsymbol{i} + s\boldsymbol{v}(x)$.

3.2.2 Oblivious Routing Scheme

As with EBS, VBS's oblivious routing scheme is based around Valiant Load Balancing (VLB). First, traffic is routed along a semi-path from the source to a random intermediate node in the network, and then traffic is routed along a second semi-path from the intermediate node to its final destination. As in EBS, the same algorithm is used to generate semi-paths in both stages of VLB. However, unlike in EBS, semi-paths are only defined starting at phase boundaries. Thus, the first step of a VBS path is to traverse up to $p-2$ virtual edges until a phase boundary is reached. Semi-paths are then defined for a given (q, a, c) triple, where the starting phase number $q = t/(p-1)$ for some timeslot t at the beginning of a phase (hence t is divisible by $p-1$). Following the initial virtual edges to reach a phase boundary, we concatenate the semi-path from the source to the intermediate node, followed by the semi-path from the intermediate node to the destination.

Depending on the current phase and the source-destination pair, we either route semi-paths via a single-basis path or a hop-efficient path. The routing scheme always selects a hop-efficient semi-path when one is available, and otherwise it selects a single-basis path. Based on a careful definition of the number of phases Q, we show how to ensure that hop-efficient paths are available a δ fraction of the time, for a parameter δ that we define later. We describe both semi-path types below.

Single-basis paths The single-basis path, or SB path, for a given (q, a, c) is formed as follows: First, we define the distance vector $\boldsymbol{d} = c - a$, as well as the basis $Y = \big(\boldsymbol{v}(q), \boldsymbol{v}(q+1), ..., \boldsymbol{v}(q+h)\big)$. Note that the vectors in the basis Y are those used to form the $h+1$ phases beginning with phase q. Then, we find the basis representation of $\boldsymbol{d}$ using basis Y, $\boldsymbol{s} = Y^{-1}\boldsymbol{d}$. Over the next $h+1$ phases, for every timeslot $t' \equiv (x', s')$, if $s' = \boldsymbol{s}_{x'}$, the physical edge is traversed. Otherwise, the virtual edge is traversed. This strategy corresponds to traversing $\boldsymbol{d}$ through its decomposition in basis Y, beginning at node a and ending at

node c.

Although this algorithm for SB paths completes within $h+1$ phases, following this virtual edges are traversed for a further Q phases. This ensures that both single-basis and hop-efficient paths (described below) each take $h+1+Q$ phases to complete. Note that it is possible for an SB path to have fewer than $h+1$ hops, although this becomes increasingly rare as N grows without bound.

Hop-efficient paths A hop-efficient path, or HE path, is formed as follows: First, for $h+1$ phases, only virtual edges are traversed. This ensures that the physical hops of HE and SB paths beginning during the same phase q use disjoint sets of vectors (assuming $p > h+1+Q$), which simplifies later analysis. Following this initial buffer period, h phases are selected out of the next Q phases, and one physical hop is taken in each selected phase. During all other timeslots within the Q phases, virtual hops are taken.

For a given starting phase q and starting node a, there are $\binom{Q}{h}(p-1)^h$ possible HE paths. As stated earlier, we will show that HE paths are available a δ fraction of the time. Because there are a total of N destinations reachable from a, we would then like δN destinations to be reachable by HE paths. Ignoring for now the possibility of destinations reachable by multiple HE paths, we set Q to the lowest integer value such that:

$$\binom{Q}{h}(p-1)^h \geq \delta N \impliedby \binom{Q}{h} \geq \delta p$$

Note that for this value of Q, $\binom{Q-1}{h} < \delta n$. For some (q, a, c), more than one HE path may exist. In this case, an arbitrary selection can be made between these multiple paths; the specific path chosen does not affect our analysis of VBS.

3.2.3 Latency-Throughput Tradeoff of VBS

Latency

A VBS path begins with at most $p-2$ virtual edges traversed until a phase boundary is reached. Following this, the first semi-path immediately begins, followed by the second semi-path. Because both SB and HE paths are defined to take $h+1+Q$ phases, the latency of a single semi-path is $(p-1)(h+1+Q)$. This gives a total maximum latency of $(p-2)+2(p-1)(h+1+Q) = (p-1)(3+2h+2Q)-1$ for VBS paths.

Throughput

Lemma 4. *Let R be the VBS routing scheme for a given N, h, and δ, such that $\delta \leq \frac{1}{4(h+1)(1+\frac{1}{2h})^2}$. For all demand functions D requesting throughput at most $\frac{1}{2(h+1-\varepsilon)}$, where $\varepsilon = \frac{1}{4}\delta$, the flow $f(R,D)$ is feasible.*

Proof. Consider an arbitrary demand function D requesting throughput at most r, and consider an arbitrary physical edge $e \in W_{\text{phys}}$ from (i,t_e) to (j,t_e+1), where t_e is the timeslot during which the edge begins. Let $t_e \equiv (x_e, s_e)$ such that x_e is the phase in the schedule corresponding to t_e, and s_e is the scale factor used during t_e. We wish to show that $F(f(R,D),e) \leq 1$.

As in our proof of the throughput of EBS (Lemma 2), we begin by inflating D into $\hat{D}'$. Similarly, we define $\hat{R}'$, the routing protocol for semi-paths, and we decompose $f(R,\hat{D}')$ into $f(\hat{R}', \hat{D}'_{a\to c})$ and $f(\hat{R}', \hat{D}'_{c\to b})$. Note that because semi-paths begin only on phase boundaries, $\hat{R}'$ in this case does not strictly follow our definition for an oblivious routing scheme. Instead, we define $\hat{R}'_{a,c,q}$ using phases q, rather than timeslots t, for the domain. The path used for $\hat{R}'_{a,c,q}$ begins during the first timeslot of phase q. This is reflective of the definitions for

semi-paths in VBS.

To generate $\hat{D}'_{a\to c}$, note that R first batches (a, c, t) triples over the $p-1$ timeslots preceding an epoch boundary, before sampling intermediate nodes uniformly. Therefore, for all (q, a, c)

$$\hat{D}'_{a\to c}(q, a, c) = \frac{1}{N} \sum_{t\in[p-1]} \sum_{u\in[N]} \hat{D}'(q(p-1) - t, a, u) = \frac{(p-1)r}{N}$$

Similarly, because semi-paths from an intermediate node to the destination always commence exactly $h + 1 + Q$ phases after the beginning of the first semi-path, we can define $\hat{D}'_{c\to b}(t, b, c)$ as follows:

$$\hat{D}'_{c\to b}(q, c, b) = \frac{1}{N} \sum_{t\in[p-1]} \sum_{u\in[N]} \hat{D}'((q-h-1-Q)(p-1) - t, u, b) = \frac{(p-1)r}{N}$$

Note that $\hat{D}'_{a\to c} = \hat{D}'_{c\to b} = \hat{D}^{ALL}$, where $\hat{D}^{ALL}$ is the uniform all-to-all demand function $\hat{D}^{ALL}(q, a, b) = \frac{(p-1)r}{N}$ for all $(q, a, b) \in \mathbb{Z} \times [N] \times [N]$. Therefore, $F(f(R, D), e) \leq 2F(f(\hat{R}', \hat{D}^{ALL}), e)$.

To calculate $F(f(\hat{R}', \hat{D}^{ALL}), e)$, we compute the number of (q, a, c) triples whose semi-paths traverse edge e. We calculate this number as follows: First, we calculate $\#_{SB}$, which represents the number of (q, a, c) triples that have an SB path that traverses edge e. Then, we calculate $\#_{missing}$, the number of such triples that have an HE path available (and thus do not traverse e). Finally, we determine $\#_{HE}$, the number of triples that traverse e using an HE path. The total flow traversing edge e is then $F(f(\hat{R}', \hat{D}^{ALL}), e) = \frac{(p-1)r}{N}(\#_{SB} - \#_{missing} + \#_{HE})$.

To find $\#_{SB}$, we use reasoning similar to that used in Lemma 2. In order for a given (q, a, c) to have an SB path that traverses edge e, the SB path for (q, a, c) must reach node (i, t), then traverse edge e. The only values of q for which this is possible are those in the

range $q_e - h \leq q \leq q_e$. For each of these q, we can generate p^h distinct (q, a, c) triples that have SB paths that traverse edge e as follows. First, select an arbitrary $\boldsymbol{s}$ such that $s_{q_e - q} = s_e$. Then, set $a = \boldsymbol{i} - \Sigma_{q'=q}^{q_e-1} s_{q'-q} \boldsymbol{v}(q')$, and $c = a + \Sigma_{q'=q}^{q+h} s_{q'-q} \boldsymbol{v}(q')$. In this case, $\boldsymbol{s}$ corresponds to a distance vector between a and c, expressed in terms of the basis used for SB paths starting in phase q. Because of how a is set, it is clear that the SB path for (q, a, c) must traverse (i, t). In addition, because $s_{q_e - q} = s_e$, the SB path will traverse edge e instead of another edge during the same phase.

For a given q, there are p^h possible values for $\boldsymbol{s}$, because all but one of its $h + 1$ elements can be set to any value in $[p]$. There are $(h + 1)$ possible values for q, giving a total of $\#_{SB} = (h + 1)p^h$.

To find $\#_{missing}$, we compare the distance vectors of (q, a, c) triples that have SB paths which traverse e with those of (q, a, c) triples that have valid HE paths. Each vector found in the overlap between these two sets corresponds to one (q, a, c) triple that contributes to $\#_{missing}$. To reason about the former set of vectors, we return to the construction of $\boldsymbol{s}$ used to find $\#_{SB}$. For a given starting phase q, each $\boldsymbol{s}$ such that $\boldsymbol{s}_{q_e - q} = s_e$ represents a distance vector that can traverse e, expressed in terms of the basis used for SB paths starting in phase q. We can construct this basis as $Y = \big(\boldsymbol{v}(q), \boldsymbol{v}(q+1), ..., \boldsymbol{v}(q+h)\big)$. For each $\boldsymbol{s}$, $\boldsymbol{d} = Y\boldsymbol{s}$ is the same distance vector expressed using the elementary basis. The range of possible distance vectors $\boldsymbol{d}$ reachable while traversing e forms D_e, an h-dimensional affine subspace of $\mathbb{F}_p^{h+1}$ that is parallel to W_e, the linear subspace spanned by the set $Y \setminus \{\boldsymbol{v}(q_e)\}$.

Next, we consider which triples have valid HE paths. For a given starting phase q, there are Q phases which are considered for forming HE paths. Let I be a set of h phase numbers chosen from these Q phases, and let $V(I)$ be the linear subspace spanned by the vectors corresponding to the phase numbers in I. There are $\binom{Q}{h}$ ways of choosing such a set I. For each possible choice, $V(I)$ forms an h-dimensional linear subspace in F_p^{h+1}, corresponding to the distance vectors reachable via HE paths using the chosen phases. (Note that $V(I)$ must

be h-dimensional because every h distinct Vandermonde vectors are linearly independent.) Because $V(I)$ and W_e are spanned by distinct sets of h Vandermonde vectors, these linear subspaces are not equivalent, implying that $V(I)$ and D_e are not parallel. Thus, $V(I) \cap D_e$ is an affine subspace with dimension $h-1$ and contains p^{h-1} distance vectors.

Some distance vectors lie in more than one such intersection. In order to avoid overcounting $\#_{missing}$, we must remove at least this many vectors from our count. Given two sets of h chosen phase numbers I and J, $V(I)$ and $V(J)$ form two different linear subspaces of $\mathbb{F}_p^{h+1}$. As linear subspaces, both I and J contain the zero vector, as does the $(h-1)$-dimensional $I \cap J$. D_e does not contain the zero vector, so $D_e \cap I \cap J$ can only be $(h-2)$-dimensional, containing p^{h-2} distance vectors. There are fewer than $\binom{Q}{h}^2$ ways of choosing two distinct sets I and J.

Thus, for a given starting q, there are fewer than $\binom{Q}{h}p^{h-1} - \binom{Q}{h}^2 p^{h-2}$ distance vectors in the overlap between D_e and the union of all possible $V(I)$. Because there are $h+1$ possibilities for the starting q, this gives the following lower bound for $\#_{missing}$:

$$\begin{aligned}
\#_{missing} &> (h+1)\left(\binom{Q}{h}p^{h-1} - \binom{Q}{h}^2 p^{h-2}\right) \\
&\geq (h+1)\left((\delta p)p^{h-1} - \left(\binom{Q-1}{h}\frac{Q}{Q-h}\right)^2 p^{h-2}\right) \\
&> (h+1)\left(\delta p^h - \left(\delta p \frac{Q}{Q-h}\right)^2 p^{h-2}\right) \\
&= (h+1)\left(\delta p^h - \delta^2 p^h \left(\frac{Q}{Q-h}\right)^2\right)
\end{aligned}$$

To find $\#_{HE}$, note that a given (q, a, c) can only traverse edge e if $q_e - h - Q \leq q < q_e - h$, since q_e must be in the set of Q phases considered for HE paths for (q, a, c). For a given q, we can construct an HE path by selecting $h-1$ additional phases from the $Q-1$ remaining phases, and then selecting one of the $p-1$ edges within that phase to traverse. Some of these

paths may lead to the same destination, causing an overcount, but it is fine to overcount $\#_{HE}$ slightly.

$$\begin{aligned}
\#_{HE} &\leq Q\binom{Q-1}{h-1}(p-1)^{h-1} \\
&= Q\binom{Q-1}{h}\frac{h}{Q-h}(p-1)^{h-1} \\
&< \delta p h \frac{Q}{Q-h}(p-1)^{h-1} \\
&< \delta h p^h \frac{Q}{Q-h}
\end{aligned}$$

Now that we have found $\#_{SB}$, $\#_{missing}$, and $\#_{HE}$, we can finally bound $F(f(R,D),e)$:

$$\begin{aligned}
F(f(R,D),e) &\leq 2F(f(\hat{R}', \hat{D}^{ALL}),e) \\
&= 2\frac{(p-1)r}{N}(\#_{SB} - \#_{missing} + \#_{HE}) \\
&< 2\frac{(p-1)r}{N}\left((h+1)p^h - (h+1)\left(\delta p^h - \delta^2 p^h \left(\frac{Q}{Q-h}\right)^2\right) + h\delta p^h \frac{Q}{Q-h}\right) \\
&= 2\frac{(p-1)r}{N}(h+1)p^h\left(1 - \left(\delta - \delta^2\left(\frac{Q}{Q-h}\right)^2\right) + \frac{h}{h+1}\delta\frac{Q}{Q-h}\right) \\
&< 2r(h+1)\left(1 - \delta\left(1 - \frac{h}{h+1}\frac{Q}{Q-h}\right) + \delta^2\left(\frac{Q}{Q-h}\right)^2\right)
\end{aligned}$$

For $Q \geq 2h^2 - h$, $\frac{Q}{Q-h} \leq \frac{h+\frac{1}{2}}{h}$. This gives:

$$\begin{aligned}
F(f(R,D),e) &< 2r(h+1)\left(1-\delta\left(1-\frac{h}{h+1}\frac{h+\frac{1}{2}}{h}\right)+\delta^2\left(\frac{h+\frac{1}{2}}{h}\right)^2\right) \\
&= 2r(h+1)\left(1-\delta\left(1-\frac{h+\frac{1}{2}}{h+1}\right)+\delta^2\left(1+\frac{1}{2h}\right)^2\right) \\
&= 2r(h+1)\left(1-\frac{1}{2}\frac{1}{h+1}\delta+\delta^2\left(1+\frac{1}{2h}\right)^2\right) \\
&= \frac{1}{2(h+1-\varepsilon)}2(h+1)\left(1-\frac{1}{2}\frac{1}{h+1}\delta+\delta^2\left(1+\frac{1}{2h}\right)^2\right) \\
&= \frac{1}{h+1-\varepsilon}\left(h+1-\frac{1}{2}\delta+(h+1)\delta^2\left(1+\frac{1}{2h}\right)^2\right) \\
&\leq \frac{1}{h+1-\varepsilon}\left(h+1-\varepsilon\right) \\
F(f(R,D),e) &< 1
\end{aligned}$$

Note that because of how we set ε and restrict δ, $\varepsilon \leq \frac{1}{2}\delta - (h+1)\delta^2(1+\frac{1}{2h})^2$. Because the amount of flow traversing any physical edge e is less than 1, the flow $f(R,D)$ is feasible. $\square$

3.2.4 Tightness Guarantees

Theorem 2. *For all $r \in (0, 1/2]$, there is a VBS design or an EBS design which guarantees throughput r and uses maximum latency*

$$L_{max} \leq O(L^*_{orn}(r,N)).$$

Proof. The VBS design of order h with parameter δ gives maximum latency $L \leq 2(h+1)(p-1) + 2Q(p-1)$ for $h = \lfloor \frac{1}{2r} \rfloor$ and $\binom{Q}{h} \geq \delta p$, as long as $\delta \leq \frac{1}{4(h+1)(1+\frac{1}{2h})^2}$. Let $\varepsilon = h + 1 - \frac{1}{2r}$, and set $\delta = 4\varepsilon$.

We chose Q such that $\binom{Q-1}{h} < \delta p$ and $\binom{Q}{h} \geq \delta p$. Then $\binom{Q}{h} < \delta p \frac{Q}{Q-h} \leq \delta \frac{h+\frac{1}{2}}{h}$, due to

$Q \geq 2h^2 - h$. Hence $Q \leq h\left(\delta p \frac{h}{h+(1/2)}\right)^{1/h}$. We upper bound the max latency of VBS in the following way.

$$\begin{aligned}
L_{max} &\leq \max\{(h+1)(p-1) + Q(p-1), (h+1)(p-1) + (2h^2-h)(p-1)\} \\
&\leq 2(h+1)(p-1) + 2h^2(p-1) + h\left(4\varepsilon p \frac{h+\frac{1}{2}}{h}\right)^{1/h}(p-1) \\
&\leq 2(h+1)p + 2h^2 p + hn(4\varepsilon p)^{1/h}\left(\frac{2h+1}{2}\right)^{1/h} \\
&\leq (h+1)[2N^{1/(h+1)} + hN^{1/(h+1)} + (4\varepsilon N)^{1/h}\left(\frac{2h+1}{2}\right)^{1/h}] \\
&\leq O(h[hN^{1/(h+1)} + (\varepsilon N)^{1/h}])
\end{aligned}$$

For sufficiently large N (determined by ε and h, both functions of r), the second term will dominate. Thus, for large N:

$$L_{max} \leq O\left(h\left[(\varepsilon N)^{1/h} + N^{1/(h+1)}\right]\right) = O\left(L^*_{orn}(r, N)\right).$$

By Lemma 4, VBS only gives a tight latency bound when $4\varepsilon = \delta \leq \frac{1}{4(h+1)(1+\frac{1}{2h})^2}$. When ε is greater than this value, we use EBS instead. By Lemma 3, EBS gives a factor C tight bound when $\varepsilon > 2\sqrt{\frac{2h}{\pi}}\left(\frac{2e}{C}\right)^h$. We check to make sure that there exists a constant C which works for all $\varepsilon > \frac{1}{4} \cdot \frac{1}{4(h+1)(1+\frac{1}{2h})^2}$

$$2\sqrt{\frac{2h}{\pi}}\left(\frac{2e}{C}\right)^h \leq \frac{1}{4}\cdot\frac{1}{4(h+1)\left(1+\frac{1}{2h}\right)^2}$$

$$\frac{2e}{C}\left(2\sqrt{\frac{2h}{\pi}}\right)^{1/h} \leq \left(\frac{1}{16(h+1)\left(1+\frac{1}{2h}\right)^2}\right)^{1/h}$$

$$C \geq 2e\left(2\sqrt{\frac{2h}{\pi}}\right)^{1/h}\left(16(h+1)\left(1+\frac{1}{2h}\right)^2\right)^{1/h}$$

$$C \geq O\left(\sqrt{h}^{1/h}\left((h+1)\left(\frac{2h+1}{2h}\right)^2\right)^{1/h}\right) = O(1)$$

Since there exists such a factor C, the following holds for EBS in the regions of interest.

$$L_{max} \leq O\left(h\left[(\varepsilon N)^{1/h} + N^{1/(h+1)}\right]\right) = O\left(L^*_{orn}(r,N)\right)$$

□

3.3 EBS and VBS for Degree $d > 1$

Recall from Section 2.2 that an upper bound for 1-regular designs will only imply a similar upper bound for d-regular designs if we can ensure that the routing scheme does not route flow paths on multiple edges in the same "unrolled" segment of the 1-degree virtual topology. EBS and VBS always route flow on paths which use at most 1 edge from each phase, where a phase constitutes $(p-1)$ timeslots. Trivially, if d divides $(p-1)$, then these constructions already have the property we need. However, even if d does not divide $(p-1)$, as long as $d < p-1$, we can modify EBS and VBS as follows.

We change the connection schedule to iterate through each phase twice before moving on to the next. So for VBS, $\pi_{(p-1)x+s-1}(\boldsymbol{i}) = \boldsymbol{i} + s\boldsymbol{v}(\lfloor x/2 \rfloor)$. We also change the definition of

single-basis and hop-efficient paths to use exclusively even-numbered phases or exclusively odd-numbered phases, depending on whether the next phase starts after the request originates. With this modification, single-basis and hop-efficient paths always use physical edges that occur at least $(p-1)$ timeslots apart from each other. Therefore, in the "rolled up" virtual topology, our flow paths will always use at most one physical edge per timeslot. This at most doubles the maximum latency, and does not affect throughput.

CHAPTER 4

EXTENDING ORN DESIGNS TO SUFFICIENTLY LARGE N

In this chapter, we prove the following extension of Theorem 1.2.

Theorem 3. *Given a guaranteed throughput value $r \in (0, 1/2]$ for which $\frac{1}{2r} \notin \mathbb{Z}$, there exists an integer N_0 such that, for all integers $N \geq N_0$, there exists an ORN design on N nodes which guarantees throughput r and achieves maximum latency within $\boldsymbol{O}\left(L^*_{orn}(r, N)\right)$.*

A major limitation of the EBS and VBS designs of Chapter 3 is that they are not defined for all network sizes. The EBS design requires the number of nodes in the network, N, to be a perfect h^{th} power, where $h = \lfloor \frac{1}{2r} \rfloor$. The VBS design is even more restrictive: it requires N to be a perfect $(h+1)^{\text{th}}$ power of a prime number. In practice the number of nodes in a datacenter network would rarely satisfy these restrictions, so the EBS and VBS designs must be regarded mainly as a theoretical exercise unless this limitation can be removed. In this chapter, we remove this limitation on the network size. We show that there exist oblivious reconfigurable network designs achieving maximum latency $\boldsymbol{O}(L^*_{orn}(r, N))$ for all sufficiently large N, whenever $\frac{1}{2r}$ is not an integer.

To shed light on the challenge of proving such a result, it helps to reflect on the constrast between network designs using N nodes and algorithm designs with input size N. In algorithm designs, it is common to define an algorithm first on special input sizes (e.g., when N is a power of two) and then to extend the algorithm to all N by "padding" the input. For example, Strassen's matrix multiplication algorithm works by reducing N-by-N matrix multiplication to seven instances of $\left(\frac{N}{2}\right)$-by-$\left(\frac{N}{2}\right)$ matrix multiplication and then solves those instances recursively. This requires N to be a power of 2; to use the same algorithm when N lies strictly between two powers of 2, one first pads the matrices with zeros until the number of rows and columns equals the next power of 2 greater than N. The padding scheme works because the extra zeros function as placeholders that have no effect on the relevant entries

of the matrix product. In networking, there is no corresponding way to "pad a network" with fictional nodes that have no effect on the actual physical nodes comprising the network. Either one assigns physical nodes to play the role of the fictional ones — but then the load on those physical nodes is affected — or one allocates no physical resources to do the work of the fictional nodes, but this affects the physical nodes of the network when they try to communicate using a routing path that goes through one of the fictional nodes.

In this chapter, we define a padding scheme that circumvents these difficulties and allows us to extend the EBS and VBS network designs to all sufficiently large values of N. To do so, we pad the network with "dummy nodes" until the total number of physical and dummy nodes matches one of the network sizes for which EBS or VBS is defined. Then, rather than allocating physical resources to do the work of the dummy nodes, we simply eliminate all of the flow on routing paths that pass through dummy nodes. In order to ensure that the resulting network design guarantees throughput r, in the padded network we use the EBS or VBS design with throughput guarantee $r/(1-\delta)$, where $\delta > 0$ is chosen to be small enough that the maximum latency increases by only a constant factor. To prove that the guaranteed throughput is indeed greater than or equal to r, we must show that in the worst case over all potential traffic demand matrices, the fraction of flow that EBS or VBS routes through dummy nodes (henceforth, "dummy flow") will not exceed δ. It turns out that this step of the analysis is sensitive to the placement of the dummy nodes, requiring us to identify a way of placing dummy nodes such that for any individual source or destination node, the total amount of dummy flow originating at that source (or terminating at that destination) is not too large.

4.1 EBS Dummy Node Design

4.1.1 Connection Schedule and Routing Scheme

Let h, ε be defined given r as before: $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. Let N be the number of nodes we would like to run Dummy EBS on. That is, N is an integer that is not an integer h-power.

Let $M \geq N$ be the smallest such M for which there exists an integer m and $M = m^h$, and consider the h-level EBS design on M nodes. By Proposition 1 and Lemma 2, this design can guarantee throughput $\frac{1}{2h}$ within max latency $(2h+1)(m-1)-1$ in a network of exactly M nodes.

We will designate a set of potential dummy nodes $\mathcal{D}$, and define the EBS dummy node design at level h on N nodes in the following way: choose a subset $D \subseteq \mathcal{D}$ such that $N + |D| = M$. All flow paths that do not travel through nodes in D remain untouched and continue to route flow as specified by EBS on M nodes. Flow paths that travel through D no longer send flow.

Note that this design will be able to guarantee a slightly smaller throughput value than previously, and some node pairs and starting timeslots may have more flow attributed to them by the routing scheme than others. To fix this second point, one can normalize the amount of flow attributed to each pair by the minimum over all pairs.

To prove what throughput value this design guarantees, we will show that if we eliminate all flow that would have been routed through nodes in $\mathcal{D}$ using EBS, no pair loses more than a δ fraction of flow for sufficiently small δ.

Designate the following set $\mathcal{D}$ as the potential dummy node set,

$$\mathcal{D} = \left\{ \left(i_0, i_1, ..., i_{h-2}, \ell + \sum_{j=0}^{h-2} i_j \right) : i_0, ..., i_{h-2} \in [m] \text{ and } \ell \in \{0, ..., h-1\} \right\}$$

Before we check that eliminating all flow on a, b, t triples that would have routed through

$\mathcal{D}$ still guarantees a high enough throughput rate, we should first check to ensure that $|\mathcal{D}|$ is large enough. That is, it must be at least $M - N$. Note that N cannot be smaller than $(m-1)^h$, as otherwise there would be a smaller integer h-power M' with $M > M' \geq N$. Therefore, it is enough to show that $|\mathcal{D}|$ is at least $m^h - (m-1)^h$.

Using the fundamental theorem of calculus, one can see that $\int_{m-1}^{m} hx^{h-1} = m^h - (m-1)^h$. The integrand hx^{h-1} will always be no more than hm^{h-1}, since x takes values in the interval $[m-1, m]$. Therefore, $\int_{m-1}^{m} hx^{h-1} \leq hm^{h-1} = |\mathcal{D}|$.

4.1.2 Tightness Guarantees

Lemma 5. *The EBS dummy node design on N nodes can guarantee throughput $\frac{1}{2h}\left(1 - \frac{2h^2}{m}\right)$ and maximum latency $(2h+1)(m-1)-1$ for $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and M equal to the smallest integer h-power larger than N, for sufficiently large N.*

Proof. We can bound the guaranteed throughput rate by bounding the amount of flow between any worst case triplet that goes through the set $\mathcal{D}$. Since EBS uses Valiant Load Balancing, we can bound the fraction of flow that gets eliminated by dummy nodes for any triple a, b, t by using the total fraction of semi-paths that get eliminated when routing semi-paths of a worst-case node a starting at timeslot t to all intermediate nodes v_{int}.

Consider node $a = (a_0, ..., a_{h-1})$ and dummy node $d = (d_0, ..., d_{h-1}) \in \mathcal{D}$, and suppose t occurs in phase x. We would like to count the number of intermediate nodes v_{int} for which the semi-path from a to v_{int} starting at timeslot t goes through d.

Suppose d and a match in all but one coordinate. Then in order for d to be on the semi-path from a to v_{int} starting at timeslot t, it must be the case that d and a are mismatched in coordinate $x+1$, and that v_{int} matched d in the $(x+1)$-th coordinate. There are m^{h-1}

such v_{int}, leaving a $\frac{1}{m}$ fraction with the property. If we choose all our dummy nodes from $\mathcal{D}$, then there are at most h dummy nodes that match a in all but the $(x+1)$-th coordinate.

Consider more generally if a d and a match in all but k consecutive coordinates, starting at phase $x+1$. If we choose all our dummy nodes from $\mathcal{D}$, then there are at most hm^{k-1} such dummy nodes. And given such a dummy node, the fraction of v_{int} that get eliminated is at most $\frac{1}{m^k}$.

Thus, if we pick all our dummy nodes from $\mathcal{D}$, we can bound the fraction of flow δ_{EBS} that gets eliminated at each node. Note that we gain a factor of 2 due to VLB, by applying this argument for both semi-paths $a \to v_{int}$ and $v_{int} \to b$.

$$\delta_{EBS} \leq 2\sum_{k=1}^{h} \frac{1}{m^k} hm^{k-1} \leq 2h \sum_{k=1}^{h} \frac{1}{m} = \frac{2h^2}{m}$$

So, we can guarantee throughput

$$r \geq \frac{1}{2h}(1-\delta_{EBS}) \geq \frac{1}{2h}\left(1 - \frac{2h^2}{m}\right).$$

□

Since the above goes toward $\frac{1}{2h}$ as $M \longrightarrow \infty$, if we wanted to guarantee a throughput value r for which $\varepsilon \neq 1$, then the EBS dummy node design on any number of nodes N can guarantee throughput r for sufficiently large N.

Lemma 6. *Let $r \in (0, \frac{1}{2})$ be a throughput value and ε, h defined as usual; $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. If $\varepsilon \in \left[\frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2}, 1\right)$, then the EBS dummy node design on N nodes achieves maximum latency $\boldsymbol{O}\left(L^*_{orn}(r,N)\right)$ for sufficiently large N.*

Proof. Let $r \in (0, \frac{1}{2})$ be given as above and let M be the smallest integer h-power larger than or equal to N.

Since $\varepsilon < 1$, then $r < \frac{1}{2h}$, meaning that for large enough N (and therefore M), $r \geq \frac{1}{2h}\left(1 - \frac{2h^2}{m}\right)$. Therefore for large enough N, the EBS dummy node scheme can guarantee throughput r and achieve maximum latency $(2h+1)(m-1)-1$.

To show that $(2h+1)(m-1)-1 \leq \mathcal{O}\left(L^*_{orn}(r,N)\right)$, we use the fact that $N \geq (m-1)^h$. Therefore $2hN^{1/h}$ is at least $2hm\left(1-\frac{h}{m}\right)^{1/h}$ which is no more than a constant factor less than $(2h+1)(m-1)-1$ for sufficiently large N.

Finally, when $\varepsilon \geq \frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2}$, by Proposition 1, the maximum latency $2hN^{1/h} \leq \mathcal{O}\left(L^*_{orn}(r,N)\right)$, completing our proof. □

4.2 VBS Dummy Node Design

The Vandermonde Basis Scheme (VBS) described in Section 3.2 is a bit trickier to work with. It requires the use of far more dummy nodes due to the primality requirement of $N^{\frac{1}{h+1}}$, and it requires some additional tweaking of the schedule to ensure that dummy nodes stay sufficiently well distributed throughout the network, due to the changing Vandermonde vector phases. Additionally, it also requires individual analysis of both single-basis and hop-efficient semi-path types.

4.2.1 Connection Schedule and Routing Scheme

Let h, ε be defined given r as before: $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. Let N be an integer that is not a prime $(h+1)$-power. That is, there is no prime number q for which $N = q^{h+1}$. Let $M > N$ be the smallest such M for which there exists a prime q and $M = q^{h+1}$. First, we define the set of possible dummy nodes $\mathcal{D}$.

$$\mathcal{D} = \left\{ \left(i_0, i_1, ..., i_{h-1}, \ell + \sum_{j=0}^{h-1} i_j \right) : i_0, ..., i_{h-1} \in [q] \text{ and } \ell \in \{0, ..., (h+1)q^{0.525} - 1\} \right\}$$

We confirm that $\mathcal{D}$ is large enough by citing the following theorem about prime gaps.

Theorem 4. [Baker, Harman, Pintz 2001] [8] *For all $x > x_0$, the interval $[x - x^\theta, x]$ contains at least one prime number for $\theta = 0.525$.*

Thus, it is sufficient to show that $|\mathcal{D}|$ is at least $M - N \leq q^{h+1} - (q - q^{0.525})^{h+1}$. Using the fundamental theorem of calculus, it is clear that

$$\begin{aligned} q^{h+1} - (q - q^{0.525})^{h+1} &= \int_{q-q^{0.525}}^{q} (h+1)x^h dx \\ &\leq (h+1)q^h(q^{0.525}) = |\mathcal{D}| \end{aligned}$$

Restricted Vandermonde Vectors In order to ensure that dummy nodes are well distributed throughout the network, we restrict the Vandermonde vectors of the VBS dummy node design so that lines parallel to these vectors intersect $\mathcal{D}$ in a limited number of points. In particular, we would like for each such line to contain no more than $(h+1)q^{0.525}$ elements of $\mathcal{D}$. In order to prove sufficiently many of these Vandermonde vectors exist, let us examine $\mathcal{D}$ more closely.

We can represent $\mathcal{D}$ as the following union

$$\begin{aligned} \mathcal{D} &= \bigcup_{k=1}^{(h+1)q^{0.525}} \mathcal{D}_k \\ &= \bigcup_{k=1}^{(h+1)q^{0.525}} \left\{ \vec{i} : i_0 + i_1 + ... + i_{h-1} + i_h = k \right\} \end{aligned}$$

We will call a Vandermonde vector *bad* if it belongs to the set $\mathcal{D}_0 = \left\{ \vec{i} : i_0 + i_1 + ... + i_{h-1} + i_h = 0 \right\}$, and otherwise we call the Vandermonde vector *good*. We will restrict our VBS

connection schedule to use only good Vandermonde vectors. Observe that the vector $\vec{v} = (1, \alpha, \alpha^2, \ldots, \alpha^h)$ is bad if and only if the equation $1 + \alpha + \alpha^2 + \ldots + \alpha^{h-1} + \alpha^h = 0$ is satisfied. This is a polynomial equation of degree h in α, so there are at most h bad Vandermonde vectors and at least $q - h$ good ones. Since the VBS connection schedule requires Q Vandermonde vectors where Q is the least integer satisfying $\binom{Q}{h} \geq 4\varepsilon q$, there are sufficiently many distinct good vectors so long as $q > h/(1 - 4\varepsilon)$.

If $\vec{v}$ is a good Vandermonde vector and L is any line parallel to $\vec{v}$, then the intersection $L \cap \mathcal{D}$ contains at most $(h+1)q^{0.525}$ elements. To see why, represent L as the set $\{\vec{a} + r \cdot \vec{v} \mid r \in \mathbb{F}_q\}$ for some $\vec{a} \in \mathbb{F}_q^{h+1}$ and recall that $\mathcal{D}$ is partitioned into the sets

$$\mathcal{D}_k = \left\{ \vec{i} \mid i_0 + \cdots + i_h = k \right\}$$

with k ranging from 1 to $(h+1)q^{0.525}$. It suffices for us to prove that $L \cap \mathcal{D}_k$ has at most one element, for each k. The relation $\vec{a} + r \cdot \vec{v} \in \mathcal{D}_k$ holds if and only if $\sum_{j=0}^{h}(a_j + r\alpha^j) = k$. Rewrite this equation as $r\left(\sum_{j=0}^{h} \alpha^j\right) = k - \sum_{j=0}^{h} a_j$, and observe that $\sum_{j=0}^{h} \alpha^j \neq 0$ because $\vec{v} = (1, \alpha, \ldots, \alpha^h)$ is a good Vandermonde vector. Hence there is a unique r satisfying the equation, and consequently $L \cap \mathcal{D}_k$ has exactly one element.

We define the VBS dummy node design at level $(h+1)$ on N nodes in the following way: let M be the smallest prime $(h+1)$-power greater than or equal to N. Choose a subset $D \subseteq \mathcal{D}$ such that $N + |D| = M$. All flow paths that do not travel through nodes in D remain untouched and continue to route flow as specified by EBS on M nodes. Flow paths that travel through D no longer send flow.

Note that in this definition, like in the EBS dummy node design, some node-timeslot triples a, b, t may have more flow attributed to them by the routing scheme than others. To fix this, one can again normalize the amount of flow attributed to each pair by the minimum over all pairs.

4.2.2 Tightness Guarantees

Lemma 7. *Let r be a throughput rate achievable by the original VBS design. Then the VBS dummy node design on N nodes can guarantee throughput at least $r\,(1-\delta_{VBS})$ for $\delta_{VBS} = \frac{2(h+1)^2}{q^{0.475}}$ and M equal to q^{h+1}, the smallest prime $(h+1)$-power larger than or equal to N.*

Proof. Consider a node $a = (a_0, ..., a_h)$ and starting timeslot t. Like in the proof of Lemma 5 we will bound the total fraction of v_{int} for which the semi-path from a to v_{int} starting at timeslot t goes through some potential dummy node, $d \in \mathcal{D}$. We will do this separately for single basis and hop efficient semi-path types.

Bounding δ for single basis paths Consider the "first worst case", when starting node a and dummy node d are exactly 1 hop apart, using one of the q hops that occur in the phase beginning next after timeslot t. Then if d is on the semi-path from (a,t) to v_{int}, it is reached on the first hop of the path. The fraction of v_{int} which may be reached on a semi-path after visiting d is $\frac{1}{q}$. The number of dummy nodes with this property with respect to a and t is at most $q^{0.525}$, by definition of the potential dummy node set $\mathcal{D}$.

More generally, we consider the "kth worst case", in which the dummy node d can be reached within the next k phases from a starting at timeslot t. In this case, the fraction of flow eliminated by d is at most $\frac{1}{q^k}$ and the amount of dummy nodes with this property is at most $(h+1)q^{0.525}q^{k-1}$.

Putting all cases together, the maximum fraction of flow that gets eliminated from single-basis semi-paths is at most

$$\delta_{SB} \leq 2\sum_{k=1}^{h+1} \frac{1}{q^k}(h+1)q^{0.525}q^{k-1}$$
$$= \frac{2(h+1)^2}{q^{0.475}} = \delta_{VBS}$$

Bounding δ for hop-efficient paths Consider a node $a = (a_0, ..., a_h)$ and timeslot t. The total number of hop-efficient semi-paths leaving (a, t) is $\binom{Q}{h}q^h$. Meanwhile, we can bound the number of semi-paths leaving (a, t) that pass through $\mathcal{D}$ in the following way: again choose the h out of Q phases that the semi-path will take hops in. If the semi-path passes through $\mathcal{D}$, then in one of the h chosen phases, the edge chosen leads to a dummy node $d \in \mathcal{D}$. Note that in each phase, there are at most $(h+1)q^{0.525}$ dummy nodes reachable by one hop. So, we can bound the number of semi-paths which leave (a, t) and pass through $\mathcal{D}$ by $\binom{Q}{h}h(h+1)q^{0.525}$. (Note that this estimation overcounts paths which pass through $\mathcal{D}$ multiple times.) So, including the factor 2 due to using 2 semi-paths per routing path, the fraction of hop-efficent path flow that gets eliminated is at most

$$\delta_{HE} \leq 2 \cdot \frac{h(h+1)q^{0.525}}{q^h}$$
$$\leq 2 \cdot \frac{(h+1)^2}{q^{0.475}} \cdot \frac{1}{q^{h-1}}$$
$$\leq \frac{2(h+1)^2}{q^{0.475}} = \delta_{VBS}$$

□

Lemma 8. *Let $r \in (0, \frac{1}{2})$ be a throughput value and ε, h defined as usual; $h = \lfloor \frac{1}{2r} \rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. If $\varepsilon \leq \frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2}$, then the VBS dummy node design can guarantee throughput r and achieve maximum latency $\boldsymbol{O}\left(L^*_{orn}(r, N)\right)$ for all sufficiently large N.*

Proof. Suppose we would like to guarantee throughput r. Then we need to find an r' such that

$r'(1-\delta_{VBS}) = r$, and show that (1) VBS works on throughput r' for large enough M, and (2) the maximum latency achieved by the VBS design on M nodes guaranteeing throughput r' is not significantly higher than $L^*_{orn}(r, N)$. That is, we'd like for $L^*_{orn}(r', M) \leq \boldsymbol{O}(L^*_{orn}(r, N))$. First, we'll examine the relationship between r' and r, and ε' and ε.

$$r'\left(1 - \frac{2(h+1)^2}{q^{0.475}}\right) = r$$
$$r' = r * \frac{q^{0.475}}{q^{0.475} - 2(h+1)^2}$$

Recall that $h = \left\lfloor \frac{1}{2r} \right\rfloor$. r' is set to be slightly larger than r, but note that it is small enough for $h' = h$. That is, $h = \left\lfloor \frac{1}{2r'} \right\rfloor$. Since r' is larger than r, then ε' will be slightly larger than ε. We can write ε' as a function of ε, h, and M below.

$$\begin{aligned}
\varepsilon' &= h + 1 - \frac{1}{2r\left(\frac{q^{0.475}}{q^{0.475} - 2(h+1)^2}\right)} \\
&= h + 1 - \frac{q^{0.475} - 2(h+1)^2}{2rq^{0.475}} \\
&= h + 1 - \frac{1}{2r} + \frac{2(h+1)^2}{q^{0.475}} \\
&= \varepsilon + \frac{2(h+1)^2}{q^{0.475}} \\
&\leq \frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2} + \frac{2(h+1)^2}{q^{0.475}} \\
&\leq \frac{1}{16(h+1)(1+\frac{1}{2h})^2}
\end{aligned}$$

The last two steps holds when we take M to be large enough. By **??**, the VBS design can guarantee throughput r' when taken with large enough M.

To show that the VBS dummy design achieves maximum latency $\boldsymbol{O}\left(L^*_{orn}(r, N)\right)$, recall that $N \geq M - (q - q^{0.525})^{h+1} \geq M - q^h(h+1)q^{0.525}$. We compare $L^*_{orn}(r', M)$ and $L^*_{orn}(r, N)$

in the following way.

$$
\begin{aligned}
& L^*_{orn}(r', M) \leq \mathcal{O}(L^*_{orn}(r, N)) \\
\Longleftarrow\ & M^{1/(h+1)} + M^{1/h}\left(\varepsilon + \frac{2(h+1)^2}{q^{0.475}}\right)^{1/h} \\
& \leq \mathcal{O}\left(\left(M\left(1 - \frac{h+1}{q^{0.475}}\right)\right)^{1/(h+1)} + \varepsilon^{1/h}\left(M\left(1 - \frac{h+1}{q^{0.475}}\right)\right)^{1/h}\right)
\end{aligned}
$$

We will separately show that the ε-related terms are a constant apart from each other, and that the M terms are a constant apart from each other (for large enough M).

For the ε-related terms, we would like for

$$\left(\varepsilon + \frac{2(h+1)^2}{q^{0.475}}\right)^{1/h} \leq \mathcal{O}(\varepsilon^{1/h})$$

Note that

$$
\begin{aligned}
& \varepsilon + \frac{2(h+1)^2}{q^{0.475}} \leq 2^h \varepsilon \\
\Longrightarrow\ & \left(\varepsilon + \frac{2(h+1)^2}{q^{0.475}}\right)^{1/h} \leq 2\varepsilon^{1/h} = \mathcal{O}(\varepsilon^{1/h})
\end{aligned}
$$

As long as M is large enough, this will hold.

Now consider the M terms,

$$M^{1/h} \text{ and } \left(M\left(1 - \frac{h+1}{q^{0.475}}\right)\right)^{1/h}.$$

To show these are a constant apart from one another, it is enough to show that for large enough M, $\left(1 - \frac{h+1}{q^{0.475}}\right)^{1/h}$ is at least a constant. This is true, as this value tends toward 1 as M goes to infinity. The same holds true for $\left(1 - \frac{h+1}{q^{0.475}}\right)^{1/(h+1)}$, found in the other relevant M term.

In total, this shows that using the VBS dummy node design, the tight throughput latency tradeoff points are achievable for any number of nodes N that is sufficiently large. $\square$

Now we have the tools to prove our main theorem, restated below.

Theorem 3. *Given a guaranteed throughput value $r \in (0, 1/2]$ for which $\frac{1}{2r} \notin \mathbb{Z}$, there exists an integer N_0 such that, for all integers $N \geq N_0$, there exists an ORN design on N nodes which guarantees throughput r and achieves maximum latency within $\boldsymbol{O}\left(L^*_{orn}(r, N)\right)$.*

Proof. Case 1: $\varepsilon \leq \frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2}$, and we use the VBS dummy node design. (Lemma 8)

Case 2: $\varepsilon \in \left(\frac{1}{2} \cdot \frac{1}{16(h+1)(1+\frac{1}{2h})^2}, 1\right)$, and we use the EBS dummy node design. (Lemma 6)

$\square$

CHAPTER 5

LOWER BOUNDS ON LATENCY

While the use of Valiant load balancing inflates path lengths by a factor of 2, which reduces throughput by a factor of 2, it turns out that this factor-2 loss is unavoidable for ORN designs. Before diving into the technical details necessary for showing that the reconfigurable network designs of Chapters 3, 4, 6 and 7 are provably optimal, we find it instructive to present a proof that no ORN design can sustain throughput greater than $\frac{1}{2} + o(1)$, even if latency is allowed to be unbounded.

Consider the following: let σ denote a random permutation of the nodes, and consider a demand function D in which every node a sends flow to destination $\sigma(a)$ at rate r. We will say a "direct link" is one whose endpoints are a and $\sigma(a)$ for some node a, and a "spraying link" is any other physical link. Define the inflated cost of a link to be 2 if it is a direct link and 1 if it is a spraying link.

This ensures that the inflated cost of *every* routing path from a to $\sigma(a)$ is at least 2, regardless of whether it is a direct or indirect path. Therefore, when an ORN design is used to route demand function D over a span of T timeslots, the total inflated cost of the links used, weighted by their flow rates, is at least $2rNT$. (In each of T timeslots, each of N nodes sends flow at rate r on a routing path of inflated cost at least 2.) On the other hand, the *expected* total inflated cost of all physical edges in the virtual topology is $\left(1 + \frac{1}{N-1}\right) NT$. This is because the virtual topology contains NT physical edges, and the expected inflated cost of each e is $1 + \frac{1}{N-1}$, accounting for the $\frac{1}{N-1}$ probability that the random permutation σ leads us to label e as a direct link and inflate its cost from 1 to 2.

If an ORN design sustains throughput r, then the flow rate on any physical edge in the virtual topology when routing demand function D is at most 1, and consequently the total inflated cost of all the physical edges used, weighted by their flow rates, is bounded

above by the combined inflated cost of all the physical edges in the virtual topology. Hence $2rNT \leq \left(1 + \frac{1}{N-1}\right) NT$ and $r \leq \frac{1}{2} + \frac{1}{2(N-1)}$. This upper bound on throughput converges to $1/2$ as $N \to \infty$.

5.1 ORN Maximum Latency

In this section we prove the lower-bound half of Theorem 1, restated below.

Theorem 1.1. *Consider any constant $r \in (0, \frac{1}{2}]$. Let (h, ε) to be the unique solution in $\mathbb{N} \times (0, 1]$ to the equation $\frac{1}{2r} = h + 1 - \varepsilon$, and let $L^*_{orn}(r, N)$ be the function*

$$L^*_{orn}(r, N) = h\left(N^{1/(h+1)} + (\varepsilon N)^{1/h}\right).$$

Then for every $N > 1$ and every ORN design on N nodes that guarantees throughput r, the maximum latency is at least $\Omega(L^*_{orn}(r, N))$.

As noted in Section 2.2, the general case of this lower bound reduces to the case $d = 1$, and we will assume $d = 1$ throughout the remainder of this section.

Because the full proof is somewhat long, we begin by sketching some of the main ideas in the proof, beginning with a much simpler argument leading to a lower bound of the form $\Omega(\frac{1}{r}N^r)$ when $1/r$ is an integer. This simple lower bound applies not only to oblivious routing schemes, but to *any* feasible flow f that solves the uniform multicommodity flow problem given by the demand function $D(t, a, b) = \frac{r}{N-1}$ for all $t \in [T]$ and $b \neq a$. The lower bound follows by combining a few key observations.

1. Define the cost of a path to be the number of physical edges it contains. Since every source sends out r units of flow at all times, the flow f sends out rNT units of flow per T-step period, in a network whose physical edges have only NT units of capacity per T-step period. Consequently the average cost of flow paths in f must be at most $\frac{1}{r}$.

2. For any source node (a, t) in the virtual topology, the number of distinct destination nodes $(b, t + L)$ that can be reached via a path with maximum latency L and cost p is bounded above by $\binom{L}{p}$.

3. If $L \leq \frac{1}{2er}N^r$, we have $\binom{L}{1/r} \leq N/4$ and $\sum_{p=1}^{1/r}\binom{L}{p} \leq N/2$, so the majority of source-destination pairs cannot be joined by a path with latency L and cost less than $\frac{1}{r} + 1$. In fact, even if we connect every source and destination with a minimum-cost path (subject to latency bound L), one can show that the average cost of paths will exceed $\frac{1}{r}$.

4. Since a feasible flow must have average path cost at most $\frac{1}{r}$, we can conclude that a feasible flow does not exist when $L \leq \frac{1}{2er}N^r$.

When $1/r$ is an integer, this lower bound of $L_{max} \geq \frac{1}{2er}N^r$ for feasible uniform multicommodity flows turns out to be tight up to a constant factor. However for oblivious routing schemes, Theorem 1.1 shows that maximum latency is bounded below by a function in which the exponent of N is roughly twice as large. Stated differently, for a given maximum latency bound, the optimal throughput guarantee for oblivious routing is only half as large as the throughput of an optimal uniform multicommodity flow.

The factor-two difference in throughput between oblivious routing and optimal uniformly multicommodity flow solutions aligns with the intuition that oblivious routing schemes must use indirect paths (as in Valiant load balancing) if they are to guarantee throughput r, whereas uniform multicommodity flow solutions (in a well-designed virtual topology) can afford to satisfy all demands using shortest-path routing. The proof of the lower bound for oblivious routing needs to substantiate this intuition.

To do so, we formulate oblivious routing as a linear program and interpret the dual variables as specifying a more refined way to measure the cost of paths. Rather than defining the cost of a path to be its number of physical edges, the duality-based proof amounts to an accounting system in which the cost of using an edge depends on the endpoints of the path

in which the edge is being used. For a parameter θ which we will set to $h+1$ (unless ε is very small, in which case we'll set $\theta = h+2$), the dual accounting system assesses the cost of an edge to be 1 if its distance from the source is less than θ, plus 1 if its distance from the destination is less than θ. Thus, the cost of an edge is doubled when it is close to both the source and the destination. The doubling has the effect of equalizing the costs of direct and indirect paths: when the distance between a source and destination is at least θ, there is no difference in cost between a shortest path and one that combines two semi-paths each composed of θ physical edges.

Viewed in this way, it is intuitive that the proof manages to show that VLB routing schemes, which construct routing paths by concatenating random semi-paths with the appropriate number of physical edges, correspond to optimal solutions of the oblivious routing LP. The difficulty in the proof lies in showing that the constructed dual solution is feasible; for this, we make use of a version of the same counting argument sketched above, that bounds the number of distinct destinations reachable from a given source under constraints on the maximum latency and the maximum number of physical edges used.

5.1.1 Full Proof

Before presenting the proof of Theorem 1.1, we formalize the counting argument we reasoned about in our proof sketch.

Lemma 9. ***(Counting Lemma)*** *If in an ORN topology, some node a can reach k other nodes in at most L timeslots using at most h physical hops per path for some integer h, then $k \leq 2\binom{L}{h}$, assuming $h \leq \frac{1}{3}L$.*

Proof. If node a can reach k other nodes in $\leq L$ timeslots using exactly h physical hops per path, then $k \leq \binom{L}{h}$. Additionally, the function $\binom{L}{h}$ grows at least exponentially in base

2 — that is, $\binom{L}{h} \geq 2\binom{L}{h-1}$ — up until $h = \frac{1}{3}L$. Therefore, the number of such k is at most $\sum_{i=1}^{h} \binom{L}{i} \leq 2\binom{L}{h}$. □

Proof. (Of Theorem 1.1.) Consider the linear program below which maximizes throughput given a maximum latency constraint, L, where we let $\mathcal{P}_L(a, b, t)$ be the set of paths from $(a, t) \to (b, t + L)$ with latency at most L.

LP

$$\begin{aligned}
&\text{maximize} && r && \\
&\text{subject to} && \textstyle\sum_{P \in \mathcal{P}_L(a,b,t)} R_{a,b,t}(P) = r && \forall a, b \in [N],\ t \in [T] \\
& && \textstyle\sum_{a \in [N]} \sum_{t=0}^{T-1} \sum_{P \in \mathcal{P}_L(a,\sigma(a),t): e \in P} R_{a,\sigma(a),t}(P) \leq 1 && \forall \sigma \in S_N, e \in E_{\text{phys}} \\
& && R_{a,b,t}(P) \geq 0 && \forall a, b \in [N],\ t \in [T],\ P \in \mathcal{P}_L(a, b, t)
\end{aligned}$$

The second set of constraints, in which the parameter σ ranges over the set S_N of all permutations of $[N]$, can be reformulated as the following set of nonlinear constraints in which the maximum is again taken over all permutations σ:

$$\max_{\sigma} \left\{ \sum_{a \in [N]} \sum_{t=0}^{T-1} \sum_{P \in \mathcal{P}_L(a,\sigma(a),t): e \in P} R_{a,\sigma(a),t}(P) \right\} \leq 1 \quad \forall e \in E_{\text{phys}}$$

Note that given an edge e, this maximization over permutations σ corresponds to maximizing over perfect bipartite matchings with edge weights defined by $w_{a,b,e} = \sum_{t=0}^{T-1} \sum_{P \in \mathcal{P}_L(a,b,t): e \in P} R_{a,b,t}(P)$. This prompts the following matching LP and its dual.

Matching LP

$$\begin{aligned}
\text{maximize} \quad & \textstyle\sum_{a,b} u_{a,b,e} w_{a,b,e} \\
\text{subject to} \quad & \textstyle\sum_{b \in [N]} u_{a,b,e} \leq 1 && \forall a \in [N] \\
& \textstyle\sum_{a \in [N]} u_{a,b,e} \leq 1 && \forall b \in [N] \\
& u_{a,b,e} \geq 0 && \forall a, b \in [N], e \in E_{\text{phys}}
\end{aligned}$$

Matching Dual

$$\begin{aligned}
\text{minimize} \quad & \textstyle\sum_{a \in [N]} \xi_{a,e} + \sum_{b \in [N]} \eta_{b,e} \\
\text{subject to} \quad & \xi_{a,e} + \eta_{b,e} \geq w_{a,b,e} && \forall a, b \in [N] \\
& \xi_{a,e} \geq 0 && \forall a \in [N], e \in E_{\text{phys}} \\
& \eta_{b,e} \geq 0 && \forall b \in [N], e \in E_{\text{phys}}
\end{aligned}$$

We then substitute finding a feasible matching dual solution into the original LP, replace the expression $w_{a,b,e}$ with its definition $\sum_{t=0}^{T-1} \sum_{P \in \mathcal{P}_L(a,b,t): e \in P} R_{a,b,t}(P)$, and take the dual again.

LP

$$\begin{aligned}
\text{maximize} \quad & r \\
\text{subject to} \quad & \textstyle\sum_{P \in \mathcal{P}_L(a,b,t)} R_{a,b,t}(P) = r && \forall a, b \in [N],\ t \in [T] \\
& \xi_{a,e} + \eta_{b,e} \geq \textstyle\sum_{t=0}^{T-1} \sum_{P \in \mathcal{P}_L(a,b,t): e \in P} R_{a,b,t}(P) && \forall a, b \in [N], e \in E_{\text{phys}} \\
& \textstyle\sum_{a \in [N]} \xi_{a,e} + \sum_{b \in [N]} \eta_{b,e} \leq 1 && \forall e \in E_{\text{phys}} \\
& \xi_{a,e} \geq 0 && \forall a \in [N], e \in E_{\text{phys}} \\
& \eta_{b,e} \geq 0 && \forall b \in [N], e \in E_{\text{phys}} \\
& R_{a,b,t}(P) \geq 0 && \forall a, b \in [N],\ t \in [T],\ P \in \mathcal{P}_L(a,b,t)
\end{aligned}$$

Dual

minimize	$\sum_e z_e$	
subject to	$\sum_{a,b,t} x_{a,b,t} \geq 1$	
	$z_e \geq \sum_b y_{a,b,e}$	$\forall a \in [N], e \in E_{\text{phys}}$
	$z_e \geq \sum_a y_{a,b,e}$	$\forall b \in [N], e \in E_{\text{phys}}$
	$\sum_{e \in P} y_{a,b,e} \geq x_{a,b,t}$	$\forall a, b \in [N],\ t \in [T],\ P \in \boldsymbol{\mathcal{P}}_L(a, b, t)$
	$y_{a,b,e}, z_e \geq 0$	$\forall a, b \in [N],\ e \in E_{\text{phys}}$

The variables $y_{a,b,e}$ can be interpreted as either edge costs we assign dependent on source-destination pairs (a, b), or demand functions designed to overload a particular edge e. We will use both interpretations, depending on if we are comparing $y_{a,b,e}$ variables to either $x_{a,b,t}$ or z_e variables respectively. According to the fourth dual constraint, the variables $x_{a,b,t}$ can be interpreted as encoding the minimum cost of a path from (a, t) to $(b, t + L)$ subject to latency bound L. According to the second and third dual constraints, the variables z_e can be interpreted as bounding the throughput requested by the demand function $D(t, a, b) = y_{a,b,e}$. We will next define the cost inflation scheme we use to set our dual variables.

Cost inflation scheme For a given node $a \in [N]$ and cutoff $\theta \in \mathbb{Z}_{>0}$, we will classify edges e according to whether they are reachable within θ physical hops of a, counting edge e as one of the hops. (In other words, one could start at node a and cross edge e using θ or fewer physical hops.) We define this value $m_\theta^+(e, a)$ as follows.

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

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

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

To understand how these values are set, consider some path P from $(a,t) \to (b,t+L)$. If we consider the m_θ^+, m_θ^- weights on the edges of P, then the first θ physical hop edges of P have weight $m_\theta^+(e,a) = 1$ and the last θ physical hop edges of P have weight $m_\theta^-(e,b) = 1$. It may be the case that some edges have both $m_\theta^+(e,a) = m_\theta^-(e,b) = 1$, if P uses fewer than 2θ physical hops. And if P uses θ or fewer physical hops, then every physical hop edge along P has weight $m_\theta^+(e,a) = m_\theta^-(e,b) = 1$. All other weights may be 0 or 1 depending on whether those edges are otherwise reachable from a or can otherwise reach b.

We start by setting $\hat{y}_{a,b,e} = m_\theta^+(e,a) + m_\theta^-(e,b)$. Also set $\hat{x}_{a,b,t} = \min_{P \in \mathcal{P}_L(a,b,t)} \{\sum_{e \in P} \hat{y}_{a,b,e}\}$. Note that by definition, $\hat{x}$ and $\hat{y}$ variables satisfy the last dual constraint. We will next find a lower bound $w \leq \sum_{a,b,t} \hat{x}_{a,b,t}$ and use that to normalize the $\hat{x}, \hat{y}$ variables to satisfy the first dual constraint.

Note that $\sum_{e \in P} \hat{y}_{a,b,e} \geq \min\{2\theta, 2|P \cap E_{\text{phys}}|\}$. Then we can bound the sum of $\hat{x}$ variables by

$$\sum_{a,b,t} \hat{x}_{a,b,t} \geq \sum_{a,t} \sum_{b \neq a} \min_{P \in \mathcal{P}_L(a,b,t)} \{2\theta, 2|P \cap E_{\text{phys}}|\}$$

Note that $\hat{x}_{a,b,t} < 2\theta$ only when there exists some path from (a,t) to $[\![b]\!]$ which uses less than θ physical edges. We can then use the Counting Lemma to produce an upper bound on the number of $b \neq a$ which have such paths: this is at most $2\binom{L}{\theta-1}$.

So, assuming that $2\binom{L}{\theta-1} \leq N$ and that $\theta - 1 \leq L/3$, we have

$$\sum_{a,t}\sum_{b\neq a}\hat{x}_{a,b,t} \geq NT\left(2\theta\left(N-2\binom{L}{\theta-1}\right)+\binom{L}{\theta-1}\right)$$

Set

$$w = NT\left(2\theta\left(N-2\binom{L}{\theta-1}\right)+\binom{L}{\theta-1}\right),$$

and then set $y_{a,b,e} = \frac{1}{w}\hat{y}_{a,b,e}$ and $x_{a,b,t} = \frac{1}{w}\hat{x}_{a,b,t}$.

Next, we set $z_e = \max_{a,b}\{\sum_a y_{a,b,e}, \sum_b y_{a,b,e}\}$. By construction, the values of $x_{a,b,t}, y_{a,b,e}, z_e$ that we have defined satisfy the dual constraints. Then to bound throughput from above, we upper bound the sums $\sum_a y_{a,b,e}$ and $\sum_b y_{a,b,e}$, thus upper bounding the sum of z_e's.

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

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

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

Recalling that $z_e = \max_{a,b}\{\sum_a y_{a,b,e}, \sum_b y_{a,b,e}\}$, we deduce that

$$z_e \leq \frac{1}{w}\left(N-1+2\binom{L}{\theta-1}\right).$$

Using this upper bound on z_e, we find that the optimal value of the dual objective — hence also the optimal value of the primal, i.e. the maximum throughput of oblivious routing

schemes — is bounded by

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

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

$$\begin{aligned}
r-\frac{1}{2\theta} &\leq \frac{4L^{\theta-1}}{2\theta(N(\theta-1)!-2L^{\theta-1})} \\
\left(r-\frac{1}{2\theta}\right)(2\theta N(\theta-1)!)-\left(r-\frac{1}{2\theta}\right)4\theta L^{\theta-1} &\leq 4L^{\theta-1} \\
\left(r-\frac{1}{2\theta}\right)2\theta N(\theta-1)! &\leq L^{\theta-1}\left(4+\left(r-\frac{1}{2\theta}\right)4\theta\right) \\
\frac{(r-\frac{1}{2\theta})2\theta N(\theta-1)!}{4+(r-\frac{1}{2\theta})4\theta} &\leq L^{\theta-1} \\
\left(\frac{(r-\frac{1}{2\theta})2\theta N(\theta-1)!}{4+(r-\frac{1}{2\theta})4\theta}\right)^{\frac{1}{\theta-1}} &\leq L
\end{aligned}$$

Now that we have a closed form, we simplify. We use Stirling's approximation, in the form $(k!)^{\frac{1}{k}} \geq \frac{k}{e}\sqrt{2\pi k}^{\frac{1}{k}}$.

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

To set the parameter θ, first note that the above bound is positive when $r > \frac{1}{2\theta}$. Additionally, we would like to set θ as large as possible, and θ must be an integer value (otherwise the Counting Lemma doesn't make sense). Taking this into account, we set $\theta = \left\lfloor \frac{1}{2r} \right\rfloor + 1$, the nearest integer for which $(r - \frac{1}{2\theta})$ produces a positive value.

To simplify our lower bound further, let $h = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon = h + 1 - \frac{1}{2r}$. These can be interpreted in the following way: h represents the largest number of physical hops we take per path (approximately), and ε is directly related to how many pairs take paths using h physical hops instead of paths using fewer than h physical hops. Note that $\varepsilon \in (0, 1]$. This gives the restated bound below.

$$
\begin{aligned}
L &\geq \frac{h}{e}N^{1/h}\left(\frac{(r-\frac{1}{2(h+1)})(h+1)\sqrt{\frac{\pi h}{2}}}{(h+1)r+\frac{1}{2}}\right)^{1/h} \\
&= \frac{h}{e}N^{1/h}\left(\frac{\left(\frac{\varepsilon}{2(h+1)(h+1-\varepsilon)}\right)(h+1)\sqrt{\frac{\pi h}{2}}}{1+\frac{\varepsilon}{2(h+1-\varepsilon)}}\right)^{1/h} \\
&= \frac{h}{e}N^{1/h}\left(\frac{\varepsilon\sqrt{\frac{\pi h}{2}}}{2(h+1-\varepsilon)+\varepsilon)}\right)^{1/h} \\
&\geq \frac{h}{e}(\varepsilon N)^{1/h}\left(\frac{\sqrt{\frac{\pi h}{2}}}{4h}\right)^{1/h} \qquad (5.1) \\
&= \frac{h}{e}(\varepsilon N)^{1/h}\cdot\Omega(1) = \Omega\left(h(\varepsilon N)^{1/h}\right)
\end{aligned}
$$

As $\varepsilon \to 0$, this bound goes toward 0, making it meaningless for extremely small values of ε. However, for such values of ε, we simply set $\theta = h + 2$ instead, which gives the following

$$L_{max} \geq \Omega\left((h+1)N^{1/(h+1)}\right)$$

To combine the two ways in which we set θ, we take the average of the two bounds. This gives the bound from our theorem statement,

$$L_{max} \geq \Omega\left(h\left[(\varepsilon N)^{1/h} + N^{1/(h+1)}\right]\right) = \Omega\left(L^*_{orn}(r,N)\right).$$

□

5.2 ORN Maximum Latency With High Probability

In this section we prove the following theorem concerning ORN designs which achieve their throughput value only with high probability.

Theorem 5. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and let*

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

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

Proof. We will start by upper bounding throughput for a given maximum latency. We begin with a set of $N!$ linear programs, one for each possible permutation σ on the node set, to solve the following problem: given maximum latency L and some reconfigurable network

schedule, LP_σ finds a set of routing paths to route r flow between each $a, \sigma(a)$ pair, and maximizes the value r for which this is possible.

Primal LP_σ

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

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

Dual_σ

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

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

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

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

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

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

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

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

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

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

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

Dual_σ solutions to bound general throughput. Now we'll show good dual solutions

for general r. Given parameter $\theta \in \mathbb{Z}_{\geq 1}$, set

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

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

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

To bound the above value, we apply the Counting Lemma from Section 5.1.1, restated below.

Lemma 9. ***(Counting Lemma)*** *If in an ORN topology, some node a can reach k other nodes in at most L timesteps using at most h physical hops per path for some integer h, then $k \leq 2\binom{L}{h}$, assuming $h \leq \frac{1}{3}L$.*

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

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

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

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

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

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

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

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

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

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

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

$$L \geq \frac{g}{2e}\left(\varepsilon N\right)^{1/g}\left(\frac{\sqrt{2\pi g}}{4(h+1)(g+2-\varepsilon)}\right)^{1/g}$$

$$\implies L \geq \Omega\left(g\left((\varepsilon N)^{1/g}+N^{1/(g+1)}\right)\right).$$

□

5.2.1 SORN Maximum Latency

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

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

Before we begin the proof, note that this lower bound does not make any claims about what maximum latencies are achievable with high probability for SORNs which guarantee throughput r. In Section 5.4, we give a similar lower bound on the *average* (or, expected) latency of any SORN design which guarantees throughput r. This lower bound has an additional multiplicative dependence on ε. Thus, the lower bound on maximum latency and the lower bound on expected latency match to within a constant factor for most values of r: when $\frac{1}{r} \notin \bigcup_{m=2}^{\infty} \left(m - \frac{1}{K}, m\right)$, for any large constant K.

Proof. The linear program appearing in Section 5.2, Theorem 5, when considered as whole instead of as a family of $N!$ different programs, sets up this SORN problem exactly. It asks: given a particular reconfigurable network schedule, for each possible permutation σ, maximize the guaranteed throughput rate while routing flow between σ-matched pairs. Since the expectation $\mathbb{E}_{\sigma}\left[\sum_e \beta_{\sigma e}\right]$, is upper bound by $NT\left(1 + \frac{4\theta}{N}\binom{2L}{\theta-1}\right)$, then there exists at least one σ for which the bound holds. The rest of the proof follows similarly to the proof of Theorem 5, only without the factor $\left(1 - \frac{1}{N^d}\right)$.

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

□

5.3 ORN Average Latency

We devote this section to proving Theorem 14 as stated in Section 7.4, and also stated below.

Theorem 14. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $h = h(r) = \left\lfloor \frac{1}{2r} \right\rfloor$ and $\varepsilon_o = \varepsilon_o(r) =$*

$h + 1 - \frac{1}{2r}$, *and let* $L_{obl}(r, N)$ *be the function*

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

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

Proof. We begin by showing that the average latency of any fixed ORN design which guarantees throughput r with respect to time-stationary demands must satisfy average latency at least $\Omega(L_{orn}(r, N))$. This will be enough to proof Theorem 14. Note that every ORN design $\mathcal{R}$ within the support of $\mathscr{R}$ must guarantee throughput r (with probability 1). Thus, each design $\mathcal{R}$ must satisfy average latency at least $\Omega(L_{orn}(r, N))$. Use linearity of expectation to then show that the expected average latency of $\mathcal{R} \sim \mathscr{R}$ must be at least $\Omega(L_{orn}(r, N))$.

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

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

Primal LP

maximize r

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

Where we interpret $lat(P)$ as the latency of the path P, or the combined number of virtual and physical edges[1]. As in Section 5.1, we replace the factorial number of constraints ranging over choices of σ with a polynomial number of constraints which range over choices of $a, b \in [N]$. We do this by interpreting these constraints for a fixed edge e as solving a maximum bipartite matching problem from $[N]$ to $[N]$. See Section 5.1.1 for a step-by-step explanation.

Primal LP

maximize r

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

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

Dual

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

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

We define this value $m^+_\theta(e,a)$ as follows, parameterized by some parameter k (to be defined later).

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

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

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

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

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

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

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

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

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

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

We use the following lemma, restated from Section 5.1.1, to bound $\sum_{a,b,t} \hat{x}_{abt}$.

Lemma 9: ***(Counting Lemma)*** *If in an ORN topology, some node a can reach k other nodes in at most L timesteps using at most h physical hops per path for some integer h, then $k \leq 2\binom{L}{h}$, assuming $h \leq \frac{1}{3}L$.*

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

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

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

and $\sum_b y_{a,b,e}$, allowing us to upper bound the total sum of z_e variables.

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

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

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

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

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

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

$$kL \geq \left(\frac{\left(r - \frac{k}{2\theta(k-1)}\right) 2\theta \frac{k-1}{k} N(\theta-1)!}{4\left(1 + \theta\left(r - \frac{k}{2\theta(k-1)}\right)\right)} \right)^{\frac{1}{\theta-1}}$$

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

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

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

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

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

to guaranteed throughput r. Let $r' = \frac{1}{2(h+1)}$. Then $r' < r$. Additionally,

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

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

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

□

5.4 SORN Average Latency

Theorem 6. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $g = g(r) = \left\lfloor \frac{1}{r} - 1 \right\rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and let $L_{so}(r, N)$ be the function*

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

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

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

Primal LP

maximize r

subject to $\sum_{P \in \mathcal{P}(a,\sigma(a),t)} S_\sigma(a,t,P) = r \qquad \forall a \in [N], t \in [T], \sigma \in S_N$

$\sum_{a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t): e \in P} S_\sigma(a,t,P) \leq 1 \qquad \forall e \in E_{\text{phys}}, \sigma \in S_N$

$\sum_{a,t} \sum_{P \in \mathcal{P}(a,\sigma(a),t): e \in P} S_\sigma(a,t,P) \cdot lat(P) \leq r \cdot NTLN!$

$S_\sigma(a,t,P) \geq 0 \qquad \forall a \in [N], t \in [T], \sigma \in S_N, P \in \boldsymbol{\mathcal{P}}(a,\sigma(a),t)$

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

Dual$_\sigma$

minimize $\sum_{e,\sigma} \beta_{\sigma e}$

subject to $\sum_{a,t,\sigma} \alpha_{at\sigma} - \gamma NTLN! \geq 1$

$\alpha_{at\sigma} \leq \sum_{e \in P} \beta_{\sigma e} + \gamma \cdot lat(P) \quad \forall a \in [N], t \in [T], \sigma \in S_N, P \in \boldsymbol{\mathcal{P}}(a,\sigma(a),t)$

$\gamma, \beta_{\sigma e} \geq 0 \qquad \forall e \in E_{\text{phys}}$

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

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

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

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

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

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

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

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

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

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

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

This is a sum over the number of physical hops m taken before edge e. For each value m, we multiply the number of nodes a which can reach edge e using m physical hops in latency

no more than kL by $\frac{1}{N}$ times the number of nodes b reachable from e using the remaining $(\theta - 1 - m)$ physical hops in latency no more than kL. Then

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

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

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

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

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

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

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

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

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

$$\implies L \geq \Omega\left(\varepsilon(\varepsilon N)^{1/g}\right).$$

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

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

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

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

□

CHAPTER 6

ORN DESIGNS WITH HIGH PROBABILITY

In this chapter, we show that the ability to *randomize the network topology* in reconfigurable networks allows oblivious routing schemes that improve upon VLB. We obtain reconfigurable network designs that improve upon the maximum latency achievable for a given throughput value by nearly the square root, when the network is allowed a small probability of violating the throughput guarantee.

This chapter is dedicated to proving the following theorem.

Theorem 7. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and let*

$$L^*_{upp}(r, N) = gN^{1/g} \tag{6.1}$$

$$L^*_{low}(r, N) = g\left((\varepsilon N)^{1/g} + N^{1/(g+1)}\right) \tag{6.2}$$

Assuming $\varepsilon \neq 1$:

1. *for infinitely many network sizes N, there exists a single, fixed ORN design that attains maximum latency $\tilde{\boldsymbol{O}}(L^*_{upp}(r, N))$, and achieves throughput r with high probability over the uniform distribution on permutation demands;*
2. *for infinitely many network sizes N, there exists a family of* distributions *over ORN designs which attains maximum latency $\tilde{\boldsymbol{O}}(L^*_{upp}(r, N))$, and achieves throughput r with high probability;*
3. *furthermore, any fixed ORN design $\boldsymbol{\mathcal{R}}$ of size N which achieves throughput r with high probability over time-stationary demands must suffer at least $\Omega(L^*_{low}(r, N))$ maximum latency.*

The upper and lower bounds on lines (6.1)-(6.2) match to within a constant factor for most values of r: when $\frac{1}{r} \notin \bigcup_{m=2}^{\infty} \left(m - \frac{2}{2^m}, m\right]$ then $\varepsilon \geq 2^{-g}$, so $L^*_{low} \geq \frac{1}{2} L^*_{upp}$. The latency of our reconfigurable network designs is $L^*_{upp} \cdot \tilde{O}(\log N)$, hence the upper and lower bounds in Theorem 7 agree within a $\tilde{O}(\log N)$ factor for most values of r. See Figure 6.1 for a visualization of these bounds. Additionally, like in Chapter 4, we condition against $\varepsilon = 1$. This is due to requiring a strictly positive slack factor between the throughput r and $\frac{1}{g+1}$.

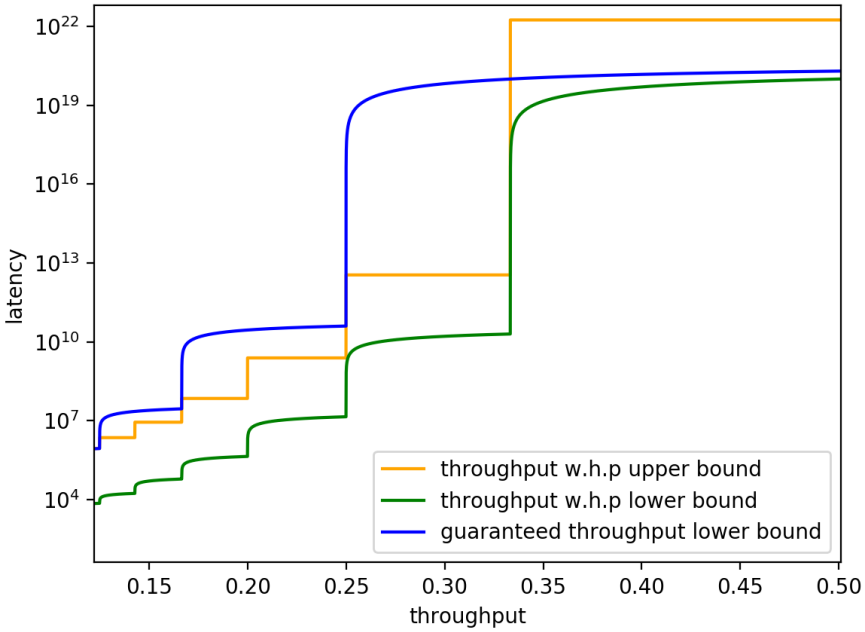

Figure 6.1: Throughput versus log-scale maximum latency tradeoff curves $\tilde{O}(L^*_{upp})$ and L^*_{low}, when compared against L^*_{orn}, the optimal tradeoff curve for guaranteed throughput from Chapters 3 and 4 and Section 5.1, on an ORN containing 10^{30} nodes.

Note that Theorem 7.3, the lower bound, is just a restatement of Theorem 5, the proof of which can be found in Chapter 5, Section 5.2.

We will begin by proving Theorem 7.1. That is, we will construct a single ORN design $\mathcal{R}^0$ which is parameterized by N, g, and C, where C is a parameter which we set during our analysis to a suitable function of N and r designed to achieve the appropriate tradeoffs between throughput and latency, and we will show that this design satisfies Theorem 7.1. We will then analyze $\mathscr{R}_N(g, C)$, a distribution over all ORN designs $\mathcal{R}^\tau$ which are equivalent to $\mathcal{R}^0$ up to re-labeling of nodes, and show that it satisfies Theorem 7.2.

6.1 Connection Schedule

The connection schedule of $\mathcal{R}^0$, like the Vandermonde Basis Scheme from Section 3.2, is based on round-robin phases (cf. Definition 13) defined by Vandermonde vectors. We interpret the set of nodes as elements of the vector space $\mathbb{F}_p^g$ over the prime field $\mathbb{F}_p$, where $N = p^g$. Each node $a \in [N]$ can then be interpreted as a unique g-tuple $(a_1, a_2, \ldots, a_g) \in \mathbb{F}_p^g$.

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

More formally, we identify each congruence class $k \pmod{T}$ with a phase number x and a scale factor s, $0 \leq x < p$ and $1 \leq s < p$, such that $k = (p-1)x + s - 1$. It is useful to think of timesteps as being indexed by ordered pairs (x, s) rather than by the corresponding congruence class mod T, so we will sometimes abuse notation and refer to timestep (x, s), when we mean timestep $k = (p-1)x + s - 1$. The connection schedule of $\mathcal{R}^0$, during timesteps $t \equiv k \pmod{T}$, uses permutation $\pi_k^0(a) = a + s\boldsymbol{v}(x)$, where x and s are the phase number and scale associated to k. Thus, each phase takes $(p-1)$ timesteps, and allows each node a to connect with nodes a' where the difference $a' - a$ belongs to the one-dimensional linear subspace generated by $\boldsymbol{v}(x)$.

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

6.2 Routing Scheme

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

Definition 14. A k-hop *pseudo-path* from a to b starting at time t is a sequence of ordered pairs $(x_1, \alpha_1), \ldots, (x_k, \alpha_k)$ such that:

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

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

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

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

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

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

$$b - a = \alpha_1 \boldsymbol{v}(x_1) + \cdots + \alpha_{g+1} \boldsymbol{v}(x_{g+1}) \tag{6.3}$$

that satisfy $\alpha_1 \neq 0$, $\alpha_{g+1} \neq 0$. For any $i \in [g+1]$ and $\beta \in \mathbb{F}_p$, there is a unique solution to (6.3) with $\alpha_i = \beta$. This is because the equation

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

is a system of g linear equations in g unknowns, with an invertible coefficient matrix. (Here we have used the fact that the vectors $\boldsymbol{v}(x_j)$ are distinct Vandermonde vectors, hence linearly independent.) Hence, the total number of solutions of (6.3) is p, and there is exactly one solution with $\alpha_1 = 0$ and exactly one solution with $\alpha_{g+1} = 0$. The number of solutions with $\alpha_1 \neq 0$ and $\alpha_{g+1} \neq 0$ is therefore either $p-2$ or $p-1$. Since there are C^{g+1} ways to choose the

$g+1$ distinct phases $x_1, \ldots, x_{g+1}$, we conclude that the number of non-degenerate $(g+1)$-hop pseudo-paths from a to b starting at time t is between $(p-2)C^{g+1}$ and $(p-1)C^{g+1}$.

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

6.3 Latency-Throughput Tradeoff

It is clear that any design $\mathcal{R}^\tau \sim \mathscr{R}_N(g, C)$ will have maximum latency $C(g+2)(p-1) < C(g+2)N^{1/g}$. (The factor of $g+2$ reflects the fact that messages wait for the duration of at most one phase block, then use the following $g+1$ phase blocks to reach their destination.) Thus, we focus on proving the achieved throughput rate with high probability in this section. Parts 1 and 2 of the following theorem correspond to parts 1 and 2 of Theorem 7, respectively.

Theorem 8. *Given a fixed throughput value r, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and assume $\varepsilon \neq 1$. As N ranges over the set of prime powers p^g for primes p exceeding $\max\left\{C(g+1), 2 + \frac{2}{1-\varepsilon}\right\}$, let $\gamma = \ln\left(\frac{g-\varepsilon-2/(p-2)}{g-1}\right)$ and $C = \frac{\log\log N}{\gamma^2}\ln(N)$. Then:*

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

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

Both parts of Theorem 8 will be proven by focusing on the congestion of physical edges in the design $\mathcal{R}^0$. For part 1, the focus on edges in $\mathcal{R}^0$ is obvious. For part 2, we make use of

the isomorphism between $\mathcal{R}^\tau$ and $\mathcal{R}^0$. Rather than considering a fixed demand function D and random design $\mathcal{R}^\tau$, we may consider the fixed design $\mathcal{R}^0$ and random demand function $D^\tau(t) = P^{-1}D(t)P$ where P denotes the permutation matrix with $P_{i,\tau(i)} = 1$ for all i.

Now, focusing on any particular edge $e \in E_{\text{virt}}(\mathcal{R}^0)$, we bound the probability that e is overloaded by breaking down the (random) amount of flow traversing e as a sum, over $0 \le q \le g$, of the amount of flow that crosses e on the $(q+1)$-th hop of a routing path. We will describe how to interpret each of these random amounts of flow as the value of a bilinear form on a pair of vectors randomly sampled from an orbit of a permutation group action. (The bilinear form is related to the demand function D, and the pair of vectors is related to the routing scheme.) We will then use a Chernoff-type bound for the values of bilinear forms on permutation group orbits, to bound the probability that the amount of $(q+1)$-th hop flow crossing e is larger than average. Finally we will impose a union bound to show the probability that any edge gets overloaded is extremely small.

Existing Chernoff-type bounds for negatively associated random variables are sufficient for the tail bound in the first part of the theorem, but not for the second part. (See Remark 1 below.) Instead, we prove the following novel tail bound for the distribution of bilinear sums on orbits of a permutation group action.

Theorem 9. *Suppose $\boldsymbol{u}, \boldsymbol{v} \in (\mathbb{R}_{\ge 0})^N$ are non-zero, non-negative vectors satisfying*

$$\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \;\ge\; CN \tag{6.4}$$

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

$$B(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \ne j} D_{ij} x_i y_j. \tag{6.5}$$

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

1. *for any* $\gamma > 0$,

$$\Pr\left(B(\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq Me^{-\frac{1}{2}\gamma^2 C}; \tag{6.6}$$

2. *for any* $\gamma > 0$,

$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq 15Me^{-\frac{1}{100}\gamma^2 C}. \tag{6.7}$$

The proof of Theorem 9 is deferred to Section 6.4.

6.3.1 Proof of Theorem 8

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

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

First we examine $q = 0$. First-hop flow traversing edge e originates at source node tail(e) during the phase block preceding the one to which e belongs. There are $C(p-1)$ time steps

[1]We number phase blocks in a flow path using the convention that phase block 1 is the first *complete* phase block in the flow path. Recall from Section 6.2 that this is also the first phase block in which it is possible that the flow is transmitted on a physical edge.

[2]Note this is a different value than if edge e is the $(q+1)$-th physical hop traversed on the path. It may be the case that in some earlier phase blocks of the path, flow may not have traversed any physical hop. If this is confusing, revisit *pseudo-paths* in Section 6.2.

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

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

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

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

$\rho_q^-(i,e) \cdot \rho_{g-q}^+(e,j)/\rho_{g+1}(i,j)$. Hence,

$$\begin{aligned} X_i &= \sum_{j\in[N],\, j\neq i} \frac{\rho_q^-(i,e)\cdot\rho_{g-q}^+(e,j)}{\rho_{g+1}(i,j)} \cdot \left(\sum_{t\in I} D(t)_{ij}\right) \\ &\leq \sum_{j\in[N],\, j\neq i} \frac{\rho_q^-(i,e)\cdot\rho_{g-q}^+(e,j)\cdot rC(p-1)\cdot \overline{D}_{ij}}{(p-2)C^{g+1}} \\ &= \left(\frac{p-1}{p-2}\right) r \sum_{j\in[N],\, j\neq i} \overline{D}_{ij} \left(\frac{\rho_q^-(i,e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e,j)}{C^{g-q}}\right) \\ \sum_{i=1}^N X_i &\leq \left(\frac{p-1}{p-2}\right) r \sum_{i\neq j} \overline{D}_{ij} \left(\frac{\rho_q^-(i,e)}{C^q}\right)\left(\frac{\rho_{g-q}^+(e,j)}{C^{g-q}}\right) = \sum_{i\neq j} \overline{D}_{ij} u_i v_j \end{aligned} \tag{6.8}$$

where

$$u_i = \left(\frac{p-1}{p-2}\right) r \left(\frac{\rho_q^-(i,e)}{C^q}\right), \qquad v_j = \frac{\rho_{g-q}^+(e,j)}{C^{g-q}}. \tag{6.9}$$

To prove Theorem 8.1, when the ORN design is fixed to be $\mathcal{R}^0$ and the demand function is the time-stationary demand D_σ for a random permutation σ, then

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

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

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

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

Similarly, to prove Theorem 8.2, recall that we are drawing a random ORN design $\mathcal{R}^\tau$ from the distribution $\mathscr{R}_N(C,r)$, and that the induced $(q+1)$-th hop flow on the edge of $\mathcal{R}^\tau$ corresponding to e, under demand function D, is equal to the induced $(q+1)$-th hop flow on edge e under demand function $P^{-1}DP$. Again letting P denote the permutation matrix

representing τ, this induced flow is bounded above by

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

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

Hence, we are in a position to prove tail bounds on the induced $(q+1)$-th hop flow on edge e, using the Chernoff-type bounds in Theorem 9, provided we can estimate the norms $\|\boldsymbol{u}\|_1$, $\|\boldsymbol{v}\|_1$, $\|\boldsymbol{u}\|_\infty$, $\|\boldsymbol{v}\|_\infty$. For $\|\boldsymbol{u}\|_1$ we have $\|\boldsymbol{u}\|_1 = \frac{p-1}{p-2} \cdot \frac{r}{C^q} \cdot \sum_{i=1}^{N} \rho_q^-(i, e)$. The sum on the right side can be calculated by realizing that it counts the total number of q-hop pseudo-paths with non-zero first coefficient that end at tail(e). There are C^q ways of choosing a q-tuple of phases from the q phase blocks preceding the phase block containing e, for each such choice there are $(p-1)p^{q-1}$ ways to choose a sequence of coefficients beginning with a non-zero value. Hence,

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

Similarly,

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

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

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

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

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

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

for $p \geq 5$. If we observe that $\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N} = \frac{(p-1)^3}{p^2(p-2)} r < r$, then we may use Theorem 9 to conclude that for any $\gamma > 0$,

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

Supposing $C \geq \frac{\log\log N}{\gamma^2} \ln(N)$ for some positive integer, then we union bound over all

$C(p-1)(g+1)N$ edges of the virtual topology and all $1 \leq q \leq g-1$ to find

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

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

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

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

□

6.4 A Tail Bound for Bilinear Sums

In Section 6.3, our analysis of the distribution of the amount of flow traversing an edge e depends on certain tail bounds for the distribution of bilinear sums on orbits of a permutation group action. The relevant tail bound is stated as Theorem 9 above. This section is devoted to proving the theorem. The proof will make use of a Chernoff-type concentration bound for negatively associated random variables. We begin by recalling some definitions and facts about negative association; see [18, 27, 48] for an introduction to this topic.

Definition 15 ([27, 30])**.** A set of random variables $X_1, ..., X_n$ are *negatively associated* if for any two functions $f, g : \mathbb{R}^n \to \mathbb{R}$ that are either both monotone increasing or both monotone decreasing, and dependent on [3] disjoint subsets of indices $S_f, S_g \subseteq [n]$, then

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

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

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

Lemma 10. *Suppose A is an n by n matrix, and $X_1, ..., X_n$ are random variables sampled by the following process: sample a permutation $\pi : [n] \to [n]$ uniformly at random, and set $X_i = A[i, \pi(i)]$. If the entries of A are non-negative and A has consistently ordered rows, then $X_1, ..., X_n$ are negatively associated.*

Proof. This will be proved by induction on n. Note that negative association amounts to showing that the covariance $Cov(f(\vec{X}), g(\vec{X})) \leq 0$. Without loss of generality, since A has consistently ordered rows, we can assume that $A[i, 1] \leq ..., A[i, n]$ for all $i \in [n]$.

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

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

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

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

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

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

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

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

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

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

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

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

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

Corollary 2. *If $\boldsymbol{u}$, $\boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-negative vectors, then the random variables $X_1, X_2, \ldots, X_N$*

defined by sampling a uniformly random permutation $\pi : [N] \to [N]$ *and setting* $X_i = u_i v_{\pi(i)}$ *are negatively associated.*

Proof. The matrix $A = \boldsymbol{u}\boldsymbol{v}^\mathsf{T}$ has non-negative entries and consistently ordered rows, so we may apply Lemma 10 to deduce the corollary. □

Corollary 3. *Let* $\mathscr{X} = \{x_1, x_2, \ldots, x_m\}$ *be any multiset of non-negative numbers, and for some* $n \leq m$ *let* $X_1, X_2, \ldots, X_n$ *denote random variables obtained by drawing* n *uniformly random samples* without replacement *from* $\mathscr{X}$*. (In other words, the conditional distribution of* X_i *given* $X_1, \ldots, X_{i-1}$ *is uniform over the multiset* $\mathscr{X} \setminus \{X_1, \ldots, X_{i-1}\}$*.) Then* $X_1, \ldots, X_n$ *are negatively associated.*

Proof. The special case $n = m$, in which the variables $X_1, \ldots, X_m$ constitute a random permutation of the elements of $\mathscr{X}$, can be obtained from Corollary 2 by setting $\boldsymbol{u} = (x_1, x_2, \ldots, x_m)^\mathsf{T}$ and $\boldsymbol{v} = (1, 1, \ldots, 1)^\mathsf{T}$. The general case in which $n \leq m$ can then be obtained by observing that the property of negative association is preserved under taking subsets of a set of random variables. □

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

Lemma 11. *Suppose* $X_1, ..., X_N$ *are negatively associated variables for which* $X_i \in [0, 1]$ *always, and* $\mathbb{E}[\sum_i X_i] = \mu$*. Then Chernoff's multiplicative tail bound holds. That is, for any* $\gamma > 0$*,*

$$\Pr\left[\sum_i X_i \geq e^\gamma \mu\right] \leq \left[\exp\left(e^\gamma - 1 - \gamma e^\gamma\right)\right]^\mu < e^{-\frac{1}{2}\gamma^2 \mu} \tag{6.10}$$

$$\Pr\left[\sum_i X_i \leq e^{-\gamma} \mu\right] \leq \left[\exp\left(e^{-\gamma} - 1 + \gamma e^{-\gamma}\right)\right]^\mu. \tag{6.11}$$

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

$$\Pr\left[\sum_i X_i \leq e^{-\gamma} \mu\right] \leq e^{-\frac{1}{3}\gamma^2 \mu}. \tag{6.12}$$

The Chernoff bound is often expressed in terms of the tail probabilities $\Pr\left[\sum_i X_i \geq (1+\delta)\mu\right]$ and $\Pr\left[\sum_i X_i \leq (1-\delta)\mu\right]$, with the bound on the right side of the inequality then being written as a function of δ. For a proof, see [18, 48]. The version of the Chernoff bound stated above is obtained from the usual one by substituting $\gamma = \ln(1+\delta)$ in the first inequality and $\gamma = -\ln(1-\delta)$ in the second.

The inequality $-e^\gamma + 1 + \gamma e^\gamma \geq \frac{1}{2}\gamma^2$ is derived by writing it in the equivalent form $\int_0^\gamma te^t\,dt \geq \int_0^\gamma t\,dt$ and comparing integrands. The inequality $-e^{-\gamma} + 1 - \gamma e^{-\gamma} \geq \frac{1}{3}\gamma^2$ is justified by using Taylor's Theorem to deduce that the left side is bounded below by $\frac{1}{2}\gamma^2 - \frac{1}{3}\gamma^3$ for $0 < \gamma < 1$ and then noting that $\frac{1}{2}\gamma^2 \geq \frac{1}{3}\gamma^2 + \frac{1}{3}\gamma^3$ when $0 < \gamma < \frac{1}{2}$.

As a first application of Lemma 11 we can prove the first tail bound asserted in Theorem 9.

Lemma 12. *Suppose $\boldsymbol{u}$, $\boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying*

$$\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \;\geq\; CN \tag{6.4}$$

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

$$B(\boldsymbol{x},\boldsymbol{y}) = \sum_{i\neq j} D_{ij}x_iy_j. \tag{6.5}$$

Let $M = 1$ if D is a permutation matrix, and $M = N^2$ otherwise. If P is a uniformly random N-by-N permutation matrix then for any $\gamma > 0$,

$$\Pr\left(B(\boldsymbol{u},P\boldsymbol{v}) \;\geq\; e^\gamma \frac{\|\boldsymbol{u}\|_1\,\|\boldsymbol{v}\|_1}{N}\right) \leq Me^{-\frac{1}{2}\gamma^2 C}. \tag{6.6}$$

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

statement. The bilinear form B is thus a convex combination of at most M bilinear forms B_σ, where B_σ is defined for a permutation σ by

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

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

If τ is the random permutation such that $P_{i,\tau(i)} = 1$ for all i, then for any permutation σ the composition $\pi = \tau \circ \sigma$ is uniformly distributed over all permutations of $[N]$. Consequently, by Corollary 2, the random variables $X_i = \frac{u_i v_{\pi(i)}}{\|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty}$ are negatively associated. By construction, they take values between 0 and 1. Furthermore, the expected value of $\sum_{i=1}^N X_i$ can be computed by linearity of expectation, using the fact that the event $\pi(i) = j$ has probability $\frac{1}{N}$ for all j.

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

Applying Lemma 11, the probability that $\sum_{i=1}^N X_i$ exceeds $\frac{e^\gamma}{N}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{\|\boldsymbol{u}\|_\infty \boldsymbol{u}\|_\infty}$ is less than $e^{-(1/2)\gamma^2 C}$. Inequality (6.6) follows because $B_\sigma(\boldsymbol{u}, \boldsymbol{v}) \leq \|\boldsymbol{u}\|_\infty \|\boldsymbol{v}\|_\infty \sum_{i=1}^N X_i$. □

Remark 1. After seeing the proof of the tail bound (6.6), it is tempting to try proving an analogous tail bound for $B(P\boldsymbol{u}, P\boldsymbol{v})$ using the random variables $X_1, \ldots, X_N$ defined by

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

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

Despite the counterexample in Remark 1, we will still be able to prove a tail bound for $B(P\boldsymbol{u}, P\boldsymbol{v})$ using negative association and the Chernoff bound, however we will need

to pursue a more indirect strategy. We begin with the following tail bound for random submatrices of a non-negative rank-one matrix.

Lemma 4. *Suppose $\boldsymbol{u}$, $\boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying* (6.4)*. For any $K \leq N/2$ let (Q, R) denote a uniformly random sample from the set of ordered pairs of K-element subsets of $[N]$ that are disjoint from one another. Then for $0 < \gamma < 1$,*

$$\Pr\left(\sum_{i\in Q}\sum_{j\in R} u_i v_j \;\geq\; e^{\gamma}\frac{K^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1\right) \;\leq\; 2e^{-\frac{1}{8}\gamma^2 CK/N} \tag{6.13}$$

$$\Pr\left(\sum_{i\in Q}\sum_{j\in R} u_i v_j \;\leq\; e^{-\gamma}\frac{K^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1\right) \;\leq\; 2e^{-\frac{1}{12}\gamma^2 CK/N}. \tag{6.14}$$

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

$$\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty} \geq e^{\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \tag{6.15}$$

$$\sum_{j\in R}\frac{v_i}{\|\boldsymbol{v}\|_\infty} \geq e^{\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \tag{6.16}$$

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

$$\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty} \leq e^{-\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \tag{6.17}$$

$$\sum_{j\in R}\frac{v_i}{\|\boldsymbol{v}\|_\infty} \leq e^{-\gamma/2}\frac{K}{N}\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \tag{6.18}$$

is satisfied. To bound the probabilities of these events, let $X_1, X_2, \ldots, X_K$ be random variables obtained by drawing K uniformly random samples without replacement from the multiset $\{\frac{u_i}{\|\boldsymbol{u}\|_\infty} \mid 1 \leq i \leq n\}$ and observe that $X_1 + \cdots + X_K$ and $\sum_{i\in Q}\frac{u_i}{\|\boldsymbol{u}\|_\infty}$ are identically distributed. By Corollary 3 the random variables $X_1, \ldots, X_K$ are negatively associated, by construction they are $[0, 1]$-valued, and by linearity of expectation their sum has expected value $\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}$. The assumption that $\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \geq CN$, combined with the inequality $\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty} \leq N$, implies $\frac{K}{N}\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty} \geq CK/N$. The Chernoff bound now implies that the probability of

inequality (6.15) being satisfied is at most $e^{-\frac{1}{8}\gamma^2 CK/N}$, and the probability of inequality (6.17) being satisfied is at most $e^{-\frac{1}{12}\gamma^2 CK/N}$. A similar argument using K random variables drawn without replacement from the multiset $\{\frac{v_i}{\|\boldsymbol{v}\|_\infty} \mid 1 \leq i \leq n\}$ establishes that the probabilities of inequalities (6.16) and (6.18) being satisfied are bounded above by $e^{-\frac{1}{8}\gamma^2 CK/N}$ and $e^{-\frac{1}{12}\gamma^2 CK/N}$, respectively. The lemma now follows by applying the union bound. □

We are now ready to restate and prove Theorem 9.

Theorem 9. *Suppose $\boldsymbol{u}$, $\boldsymbol{v} \in (\mathbb{R}_{\geq 0})^N$ are non-zero, non-negative vectors satisfying*

$$\left(\frac{\|\boldsymbol{u}\|_1}{\|\boldsymbol{u}\|_\infty}\right)\left(\frac{\|\boldsymbol{v}\|_1}{\|\boldsymbol{v}\|_\infty}\right) \geq CN \tag{6.4}$$

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

$$B(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \neq j} D_{ij} x_i y_j. \tag{6.5}$$

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

1. *for any $\gamma > 0$,*

$$\Pr\left(B(\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq Me^{-\frac{1}{2}\gamma^2 C}; \tag{6.6}$$

2. *for any $\gamma > 0$,*

$$\Pr\left(B(P\boldsymbol{u}, P\boldsymbol{v}) \geq e^{\gamma}\frac{\|\boldsymbol{u}\|_1 \|\boldsymbol{v}\|_1}{N}\right) \leq 15Me^{-\frac{1}{100}\gamma^2 C}. \tag{6.7}$$

Proof. The first tail bound, inequality (6.6), was already proven in Lemma 12, so we turn to proving (6.7). As in the proof of (6.6), we will be using the Birkhoff-von Neumann Theorem, Carathéodory's Theorem, and the union bound to reduce to the case where the doubly

stochastic matrix D is a permutation matrix. Accordingly, for the remainder of the proof we will be focused on a fixed permutation σ and its associated bilinear form

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

and our goal will be to prove the tail bound (6.7) when $B = B_\sigma$ and $M = 1$.

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

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

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

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

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

1. *Every directed 2-cycle has a balanced 3-coloring.* If the cycle has length k, then color the i^{th} vertex of the cycle with $\chi(i) = i \pmod 3$ unless $k \equiv 1 \pmod 3$, in which case the first $k-1$ vertices of the cycle are colored using $\chi(i) = i \pmod 3$ and the last vertex is colored with the unique color that differs from both of its neighbors' colors.
2. *The disjoint union of two graphs with balanced 3-colorings also has a balanced 3-coloring.* If a balanced 3-coloring of a graph with n vertices, let us say that a color is *overused* if it is used more than $\lfloor n/3 \rfloor$ times. If graph G is the disjoint union of G_0 and G_1, each of which has a balanced 3-coloring, let k_0 and k_1 denote the number of overused colors in G_0 and G_1, respectively. If $k_0 + k_1 \leq 3$ then we can recolor G_1 if necessary so that its set of overused colors is disjoint from the set of overused colors in G_0. The union of the two colorings is then a balanced 3-coloring of G. If $k_0 + k_1 > 3$ then it must be the case that $k_0 = k_1 = 2$, in which case we can recolor G_1 if necessary so that each $q \in \{0, 1, 2\}$ is overused in at least one of G_0, G_1, and exactly one color is overused in both. The union of the two colorings is then a balanced coloring of G.

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

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

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

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

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

Define a random variable Y by

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

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

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

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

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

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

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

Next we turn to bounding the conditional probability $\Pr(\mathcal{E}_3^q \setminus \mathcal{E}_1^q \mid Z = z)$, for each value z in the support of Z. Recall that the value of Y is determined by Z, and the event $\mathcal{E}_1^q$ is

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

Assume henceforth that $z \in \mathcal{Z}_0$. Then $Y > e^{-\frac{5}{7}\gamma}\frac{K_q^2}{N^2}\|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_1$. Now, let $\boldsymbol{u}_Q$ denote the subvector of $\boldsymbol{u}$ indexed by the elements of Q, and let $\boldsymbol{v}_R$ denote the subvector of $\boldsymbol{v}$ indexed by the elements of R. We will apply Lemma 12 to this pair of vectors. Note that $\|\boldsymbol{u}_Q\|_1\|\boldsymbol{v}_R\|_1 = Y$. Hence,

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

By Observation **[O4]**, the random variable $B_\sigma^{(q)}(P\boldsymbol{u}, P\boldsymbol{v})$ can be calculated by sampling a uniformly random bijection π between Q and R and computing the sum $\sum_{i\in Q} u_i v_{\pi(i)}$. Hence, by Lemma 12,

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

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

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

Now, by the union bound, we find that

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

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

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

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

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

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

□

6.5 Proving the Topology Forms an Expander

In this section, we prove that the *emulated graph* G_{em} (see Definition 1) of the connection schedule from Section 6.1 forms an expander graph. We rely on the spectral definition of edge expansion, and thus we need to bound the second eigenvalue of the normalized adjacency matrix of G_{em}. The main proof in this section follows a similar structure to the Simple Proof of the Alon-Roichman Theorem by Zeph Landau and Alexander Russell [33], which relies on some representation theory. So, the beginning of this section will be devoted to a recap of the important tools from representation theory. We will then use Cheeger's inequality to show that G_{em} is an edge-expander.

Representation Theory Recap

A vector space V is a *Hilbert space* if it is equipped with an inner product $\langle \cdot, \cdot \rangle$ that induces a distance function on V, which in turn induces a complete metric space. A *unitary operator* U on a Hilbert space V is a bounded linear function $U : H \to H$ which is both surjective, and preserves the inner product of V. That is, $\langle Ux, Uy \rangle = \langle x, y \rangle$ for all $x, y \in V$.

Let H be a finite group, and V be some finite dimensional Hilbert space. A *representation*

ρ of H is a function $\rho : H \to U(V)$, where V is some finite-dimensional Hilbert space and $U(V)$ denotes the set of unitary operators on that Hilbert space. The dimension of ρ, d_ρ, is the dimension of V. If we choose some basis of V (say, the elementary basis), then we can associate each $\rho(g)$ with a unitary matrix $[\rho(g)]$, such that for every $g, h \in H$, $[\rho(gh)] = [\rho(g)][\rho(h)]$, using matrix multiplication.

Let us fix a representation $\rho : H \to U(V)$, and consider some subspace $W \subseteq V$. We say that W is *invariant* (with respect to ρ) if $\rho(g)W \subseteq W$ for all $g \in H$. (Note that if W is invariant w.r.t. ρ, then the restriction $\rho_W : H \to U(W)$ given by restricting each $\rho(g)$ to W is also a representation of W.) If W is non-trivial (i.e. $W \neq ,V$), then $W^\perp = \{u : \forall w \in W, \langle u, w\rangle = 0\}$ is also invariant w.r.t. ρ. When there is no non-trivial subspace that is invariant w.r.t. ρ, then we say ρ is irreducible.

If V has a non-trivial invariant subspace W, then we may decompose $V = W \oplus W^\perp$. This is the natural decomposition of the operators $\rho(g) = \rho_W(g) \oplus \rho_{W^\perp}(g)$. By repeating this process, any representation ρ of H may be decomposed into its irreducible representations, $\rho = \sigma_1 \oplus \ldots \oplus \sigma_k$.

Two representations ρ and σ are said to be equivalent if they are the same up to isometric change in basis. Any finite group H has a finite number of distinct irreducible representations (up to equivalence). We let $\hat{H}$ denote a set of representations containing one from each equivalence class.

Consider the trivial representation $\mathbf{1}$, which maps all elements of H to the identity operator on $\mathbb{C}$. Additionally, let R denote the regular representation, given by the permutation action on H itself. Specifically, let $\mathbb{C}[H] = \{\sum_g \alpha_g \cdot g : \alpha_g \in \mathbb{C}\}$ be the $|H|$-dimensional vector space of formal sums, equipped with an inner product for which $\langle g, h\rangle = 1$ exactly when $g = h$, and $\langle g, h\rangle = 0$ otherwise. Then $R : H \to U(\mathbb{C}[G])$ given by linearly extending the rule $R(g)[h] = gh$.

While the trivial representation $\mathbf{1}$ is irreducible, R is not; in fact, every irreducible representation $\rho \in \hat{H}$ appears in R with multiplicity equal to its dimension.

$$R = \bigoplus_{\rho \in \hat{H}} \underbrace{\rho \oplus \ldots \oplus \rho}_{d_\rho \text{ times}}$$

Note that by counting dimensions on either side of this equation, this gives us the following interpretation of $|H|$: $|H| = \sum_\rho d_\rho^2$.

Now let $\boldsymbol{A}(V)$ be the collection of self-adjoint linear operators on V, which is still a finite dimensional Hilbert space. For $A \in \boldsymbol{A}(V)$, let $\|A\|$ denote the operator norm of A which equals the largest absolute value obtained by an eigenvalue of A. And, let

$$P(V) = \{A \in \boldsymbol{A}(V) : \forall v, \langle Av, v\rangle \geq 0\}$$

be the cone of positive operators. This induces a partial order on $\boldsymbol{A}(V)$ by defining $A \geq B$ exactly when $A - B \in P(V)$. We interpret $B \in [A_1, A_2]$ to means that $A_1 \leq B \leq A_2$.

Proposition 2. *([2]) Let V be a Hilbert Space of dimension d and $A_1, \ldots, A_k$ be i.i.d. random variables taking values in $P(V)$ with expected value $\mathbb{E}[A_i] = M \geq \mu\mathbb{1}$, and $A_i \leq \mathbb{1}$. Then for all $\varepsilon \in \left[0, \frac{1}{2}\right]$,*

$$\Pr\left[\frac{1}{k}\sum_{i=1}^{k} A_i \in [(1-\varepsilon)M, (1+\varepsilon)M]\right] \leq 2d \cdot e^{\frac{-\varepsilon^2 \mu k}{2 \ln 2}}.$$

Definition 17. Let H be a finite group and $S \subset H$ be a set of generators for H. Then the *Cayley graph* $\boldsymbol{\mathcal{X}}(H, S)$ is the graph obtained by taking elements of H as the vertices, and including the edge (u, v) if $u^{-1}v \in S \cup S^{-1}$, where $S^{-1} = \{s^{-1} : s \in S\}$.

Note that since $S \cup S^{-1}$ is closed under inverse, then edges are symmetric, that is $u^{-1}v \in S \cup S^{-1} \iff v^{-1}u \in S \cup S^{-1}$.

Definition 18. Consider a graph G with node set V and edge set E. G is an *ε-edge expander* if for all subsets $S \subseteq V$ of size no more than $\frac{1}{2}|V|$,

$$|\{(u, v) : u \in S, v \notin S\}| \geq \varepsilon \sum_{v \in S} deg(v)$$

where $deg(v)$ is the degree of vertex v.

Definition 19. Let G be a d-regular graph with node set V and edge set E. The *normalized adjacency matrix* $\mathcal{A}(G)$ is the $|V| \times |V|$-dimensional matrix with elements

$$\mathcal{A}(G)[u,v] = \begin{cases} \frac{1}{d} & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

It is not difficult to show that $\mathcal{A}(G)$ only has eigenvalues in the range $[-1,1]$, and that its largest eigenvalue $\lambda_1(G)$ takes the value 1 exactly when G is a connected graph. For a regular graph, let $\lambda_2(G)$ denote the *second-largest* absolute value of an eigenvalue of $\mathcal{A}(G)$. That is, if we let $x_1, \ldots, x_{|V|}$ be the multiset of eigenvalues of $\mathcal{A}(G)$, with $x_1 = 1$, then $\lambda_2(G) = \max\{|x_2|, \ldots, |x_{|V|}|\}$.

Lemma 13. ***(Cheeger's Inequality).*** *If G is a d-regular graph, then then the second eigenvalue of $\mathcal{A}(G)$ lower bounds its expansion in the following way. If $\lambda_2(G) \leq \lambda$, then G must be an ε-edge expander with $\varepsilon \geq \frac{1-\lambda}{2}$.*

We now have the tools to state the main result of this section.

Theorem 10. *Let the graph G be the emulated graph of the connection schedule described in Section 6.1 with $h = 2$. That is, for integer p, let $x_1, \ldots, x_C \in \mathbb{Z}_p$ be a set of C scalars*[4] *drawn independently and uniformly at random which define Vandermonde vectors $\boldsymbol{v}(x_1), \ldots, \boldsymbol{v}(x_C) \in \mathbb{Z}_p^2$. G has nodes represented by vectors $\boldsymbol{a}_v$ in the vector space $\mathbb{Z}_p^2$, and edges $E = \{(u,v) : \boldsymbol{a}_u + \alpha \boldsymbol{v}(x_i) = \boldsymbol{a}_v$ for some x_i and $\alpha \in \mathbb{Z}_p\}$. Then G is a $\frac{1}{3}$-edge expander* with high probability[5] *over the random choices of $x_1, \ldots, x_C$, provided $C \geq \Omega\left(log(p)\right)$.*

Proof. Let $x_1, \ldots, x_C \in \mathbb{Z}_p$ be a set of C scalars, chosen independently and uniformly at random, which define Vandermonde vectors $\boldsymbol{v}(x_1), \ldots, \boldsymbol{v}(x_C) \in \mathbb{Z}_p^2$. For each $\boldsymbol{v}(x_i)$, define the

[4] In Section 6.1, the scalars $x_1, \ldots, x_C$ are required to be distinct. In this theorem, however, $x_1, \ldots, x_C$ are i.i.d. random variables that may realize to the same value.

[5] Similar to Definition 11, in this context we consider with high probability to mean that for all $d > 0$, there exists some constant C_d for which if $C = C_d \cdot \Omega\left(log(p)\right)$, then the probability is not more than $\frac{1}{p^d}$.

adjacency matrix A_i, corresponding to the edges of G that $\boldsymbol{v}(x_i)$ contributes to. That is, A_i indicates edges (u, v) where $\boldsymbol{a}_u + \alpha \boldsymbol{v}(x_i) = \boldsymbol{a}_v$ for some scalar $\alpha \in \mathbb{Z}_p$. Then the normalized adjacency matrix $\mathcal{A}$ of G is equivalent to $\frac{1}{C} \sum_{i=1}^{C} \frac{1}{p-1} A_i$.

Now consider an alternate way of defining the random variable $\mathcal{A}$. For each $i \in [C]$, let $\boldsymbol{w}_i = (y_i, z_i) \in \mathbb{Z}_p^2$ be a random element drawn uniformly from the subset of $\mathbb{Z}_p^2$ with non-zero first coordinate. Let adjacency matrices B_i indicate edges (u, v) where $\boldsymbol{a}_u + \alpha \boldsymbol{w}_i = \boldsymbol{a}_v$ for some scalar $\alpha \in \mathbb{Z}_p$. Then $\mathcal{A}$ is also equivalent to $\frac{1}{C} \sum_{i=1}^{C} \frac{1}{p-1} B_i$. We will use this interpretation of $\mathcal{A}$ moving forward.

Now consider the finite group $\mathbb{Z}_p^2$. Define the subset $S = \{\alpha(y_i, z_i) : \alpha \in \mathbb{Z}_p \text{ and } i \in [C]\}$, where $(y_i, z_i) \in \mathbb{Z}_p^2$ is chosen as above. Then G is the Cayley graph $\mathcal{X}(\mathbb{Z}_p^2, S)$. We then prove and use the following result to complete the bulk of the proof.

Theorem 11. *Let $\mathbb{Z}_p^2$ be the 2-dimensional finite group of integers modulo $p \geq 5$, $\varepsilon > \frac{1}{p-1}$, $\beta > 0$, and $k = \frac{8 \ln(2)}{\delta^2} (\beta + 1 + \ln(p^2))$ for $\delta = \frac{\varepsilon(p-1)-1}{p-2}$. Additionally, let $s_1, \ldots, s_k \in \mathbb{Z}_p^2$ be independent random variables, uniformly drawn from the set of elements with non-zero first coordinate, $\{(g_1, g_2) \in \mathbb{Z}_p^2 : g_1 \neq 0\}$. Finally, let the generating set $S = \{\alpha s_i : i \in [k] \text{ and } \alpha \in \{1, \ldots, p-1\}\}$. Then*

$$\Pr\left[\lambda_2(\mathcal{X}(\mathbb{Z}_p^2, S)) \notin \left[\frac{1-\varepsilon}{2}\mathbb{1}, \frac{1+\varepsilon}{2}\mathbb{1}\right]\right] \leq 2^{-\beta}.$$

We defer the proof of Theorem 11 to the end of this section, and for now return to the proof of Theorem 10. The results of Theorem 11 give us the following conclusion.

$$\Pr[\lambda_2(G) \geq \varepsilon] \leq 2^{-\beta},$$

for $\varepsilon > \frac{1}{p-1}$, $\beta > 0$, and $C = k = \frac{8 \ln(2)}{\delta^2} (\beta + 1 + \ln(p^2))$ for $\delta = \frac{\varepsilon(p-1)-1}{p-2}$. By Cheeger's inequality, $\lambda_2(G) \leq \varepsilon$ implies that G is a $\frac{1-\varepsilon}{2}$-edge expander. We would like for G to be a

$\frac{1}{3}$-edge expander, so we set $\varepsilon = \frac{1}{3}$. This results in a value

$$\begin{aligned} k &= \frac{8\ln(2)(3(p-2))^2}{(p-4)^2}\left(\beta + 1 + \ln(p^2)\right) \\ &\leq \boldsymbol{O}\left(\beta + \ln(p^2)\right) \quad \text{when } p \geq 5. \end{aligned}$$

If $\beta = \ell \log(p)$ for some constant ℓ, then by Theorem 11, when we can set $C = \ell \cdot \Omega\left(log(p)\right)$,

$$\Pr\left[\lambda_2(G) > 1/3\right] \leq \frac{1}{p^\ell},$$

which fulfills our definition of with high probability. □

Proof. (of Theorem 11)

Let u_i be the element within the $|\mathbb{Z}_p^2|$-dimensional vector space of formal sums corresponding to s_i's contribution to the generating set S. That is, $u_i = \frac{1}{p-1}\sum_\alpha(\alpha s_i) \in \mathbb{C}^{p^2}$. Define s to be the formal sum

$$s = \frac{1}{k}\sum_i u_i = \frac{1}{k(p-1)}\sum_{i,\alpha}(\alpha s_i) \in \mathbb{C}^{p^2}.$$

Note that the normalized adjacency matrix $\mathcal{A}$ of $\boldsymbol{\mathcal{X}}(\mathbb{Z}_p^2, S)$ is exactly $\hat{s}(R)$ when expressed in the basis $\{1 \cdot g : g \in \mathbb{Z}_p^2\}$ of $\mathbb{C}[\mathbb{Z}_p^2]$. Consider the decomposition of the regular representation R into its irreducible representations. This corresponds to an orthogonal direct sum decomposition of $\mathbb{C}[\mathbb{Z}_p^2]$ into spaces invariant under each $R(g)$. The eigenvalue 1 corresponds directly to the trivial representation $\mathbf{1}$. It suffices then, to bound the spectrum of $\hat{s}(R)$ when restricted to the non-trivial representations appearing in the decomposition. Specifically, $\lambda_2(\boldsymbol{\mathcal{X}}(\mathbb{Z}_p^2, S) = \max_{\rho \neq \mathbf{1}} \|\hat{s}(\rho)\|$.

For a formal sum u in $\mathbb{C}[\mathbb{Z}_p^2]$ and a (non-trivial) representation ρ of $\mathbb{Z}_p^2$, let $\hat{u}(\rho) = \sum_{g \in \mathbb{Z}_p^2} u_g \rho(g)$. So, $\hat{u_i}(\rho) = \frac{1}{p-1}\sum_\alpha \rho(\alpha s_i)$.

We need to understand $\mathbb{E}_{u_i}[\hat{u_i}(\rho)]$ in order to apply Proposition 2.

$$\begin{aligned}
\mathbb{E}_{u_i}\left[\hat{u_i}(\rho)\right] &= \frac{1}{p-1}\mathbb{E}\left[\sum_{\alpha}\rho(\alpha s_i)\right] \\
&= \frac{1}{p-1}\sum_{\alpha}\mathbb{E}\left[\rho(\alpha s_i)\right] \\
&= \frac{1}{1-p}(1-p)\mathbb{E}_{(g_1,g_2)\in\mathbb{Z}_p^2:g_1\neq 0}\left[\rho(g_1,g_2)\right] \\
&= \mathbb{E}_{(g_1,g_2):g_1\neq 0}\left[\rho(g_1,g_2)\right]
\end{aligned}$$

To calculate $\mathbb{E}_{(g_1,g_2):g_1\neq 0}\left[\rho(g_1,g_2)\right]$, note that $\mathbb{E}_{(g_1,g_2)}\left[\rho(g_1,g_2)\right]$ can be written as a convex combination of $\mathbb{E}_{(g_1,g_2):g_1=0}\left[\rho(g_1,g_2)\right]$ and $\mathbb{E}_{(g_1,g_2):g_1\neq 0}\left[\rho(g_1,g_2)\right]$.

First, note that $\mathbb{E}_{(g_1,g_2)}\left[\rho(g_1,g_2)\right] = 0$ always. Additionally, let us represent ρ using the tuple (a,b), in which the function $\rho\left(x,y\right) = e^{\frac{2\pi i}{p}(ax+by)}$. Then

$$\mathbb{E}_{(g_1,g_2):g_1=0}\left[\rho(g_1,g_2)\right] = \begin{cases} 1 & \text{if } b=0 \\ 0 & \text{otherwise} \end{cases}$$

We now compute the convex combination.

$$\begin{aligned}
\mathbb{E}_{(g_1,g_2)}\left[\rho(g)\right] &= \frac{1}{p^2}\sum_{(g_1,g_2)}\rho(g_1,g_2) \\
&= \frac{1}{p^2}\sum_{(g_1,g_2):g_1=0}\rho(g_1,g_2) + \frac{1}{p^2}\sum_{(g_1,g_2):g_1\neq 0}\rho(g_1,g_2) \\
&= \frac{p}{p^2}\frac{1}{p}\sum_{(g_1,g_2):g_1=0}\rho(g_1,g_2) + \frac{p^2-p}{p^2}\frac{1}{p^2-p}\sum_{(g_1,g_2):g_1\neq 0}\rho(g_1,g_2) \\
&= \frac{p}{p^2}\mathbb{E}_{(g_1,g_2):g_1=0}\left[\rho(g_1,g_2)\right] + \frac{p^2-p}{p^2}\mathbb{E}_{(g_1,g_2):g_1\neq 0}\left[\rho(g_1,g_2)\right]
\end{aligned}$$

We now have enough information to calculate $\mathbb{E}_{\hat{u_i}}\left[\hat{u_i}(\rho)\right]$.

$$\mathbb{E}_{\hat{u_i}}\left[\hat{u_i}(\rho)\right] = \left(\mathbb{E}_{(g_1,g_2)}\left[\rho(g)\right] - \frac{p}{p^2}\mathbb{E}_{(g_1,g_2):g_1=0}\left[\rho(g_1,g_2)\right]\right)\frac{p^2}{p^2-p}$$

$$\mathbb{E}_{\hat{u_i}}\left[\hat{u_i}(\rho)\right] = \begin{cases} -\frac{1}{p-1} & \text{if } b=0 \\ 0 & \text{otherwise} \end{cases}$$

In order to apply Proposition 2 and achieve our tail bound, we need for both $\|\hat{u_i}(\rho)\| \leq 1$, which is true, and for $\mathbb{E}_{\hat{u_i}}[\hat{u_i}(\rho)] \geq \mu\mathbb{1}$ for some constant $\mu > 0$. Since $\mathbb{E}_{\hat{u_i}}[\hat{u_i}(\rho)]$ equals either 0 or a slightly negative number, we define variables v_i to adjust this expectation. Let $v_i = \frac{1}{2}(1 + u_i) = \frac{1}{2}(1 + \frac{1}{p-1}\sum_\alpha \alpha s_i)$. Then $\hat{v_i} = \frac{1}{2}(1 + \frac{1}{p-1}\sum_\alpha \rho(\alpha s_i))$, and $\|\hat{v_i}(\rho)\| \leq 1$. Additionally,

$$\mathbb{E}[\hat{v_i}(\rho)] = \begin{cases} \frac{1}{2} - \frac{1}{2(p-1)} & \text{if } b = 0 \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

We would like to bound the probability that $\frac{1}{k}\sum_i \hat{v_i}(\rho) \notin [\frac{1-\varepsilon}{2}\mathbb{1}, \frac{1+\varepsilon}{2}\mathbb{1}]$ for each representation ρ. Let $\delta = \frac{\varepsilon(p-1)-1}{p-2}$, which note is a positive value because $\varepsilon > \frac{1}{p-1}$. Then regardless of if $\rho = (a, b)$ has $b = 0$ or not, the interval $\left[(1-\varepsilon)\mathbb{E}[\hat{v_i}(\rho)], (1+\varepsilon)\mathbb{E}[\hat{v_i}(\rho)]\right] \subseteq \left[\frac{1-\varepsilon}{2}\mathbb{1}, \frac{1+\varepsilon}{2}\mathbb{1}\right]$. We now apply Proposition 2.

$$\begin{aligned}
&\Pr\left[\lambda_2(\mathcal{X}(\mathbb{Z}_p^2, S)) \notin \left[\frac{1-\varepsilon}{2}\mathbb{1}, \frac{1+\varepsilon}{2}\mathbb{1}\right]\right] \\
&\leq \Pr\left[\exists\rho : \frac{1}{k}\sum_i \hat{v_i}(\rho) \notin \left[(1-\varepsilon)\mathbb{E}[\hat{v_i}(\rho)], (1+\varepsilon)\mathbb{E}[\hat{v_i}(\rho)]\right]\right] \\
&\leq 2p^2 \cdot \exp\left(\frac{-\delta^2(p-2)k}{4\ln 2(p-1)}\right) \\
&= 2p^2 \cdot \exp\left(\frac{-\delta^2(p-2)}{4\ln 2(p-1)} \cdot \frac{8\ln(2)}{\delta^2}(\beta + 1 + \ln(p^2))\right) \\
&= 2p^2 \cdot \exp\left(\frac{-2(p-2)}{p-1}(\beta + 1 + \ln(p^2))\right) \\
&\leq 2p^2 \exp\left(\frac{-2(p-2)(\beta+1)}{p-1} - \ln(p^2)\right) \\
&\leq 2\exp\left(\frac{-2(p-2)(\beta+1)}{p-1}\right) \\
&\leq 2^{-\beta} \quad \text{when } p \geq 5.
\end{aligned}$$

□

CHAPTER 7

SEMI-OBLIVIOUS RECONFIGURABLE NETWORK DESIGN

To improve the high-probability bound on throughput from Chapter 6 to a bound that holds with probability 1, we adopt a semi-oblivious routing protocol, which is a hybrid of a *primary scheme* identical to the oblivious routing protocol from Chapter 6, and a *failover scheme* (which is also oblivious), to be used in the low-probability case that the primary scheme produces an infeasible flow. The failover scheme has latency $\tilde{O}(N)$ and resembles VLB, distributing flow over two-hop paths from the source to the destination by routing through an intermediate node sampled from a nearly-uniform distribution. The challenge is to modify the connection schedule to ensure that, over a long enough period of the schedule T, enough two-hop paths exist between every source and destination. We accomplish this by using a time-varying sequence of *constellations* in place of a fixed set of $(g+1)$ phase blocks, described and used in Section 6.1. The time-varying sequence of constellations that we construct forms a sort of combinatorial design, covering every vector with non-zero coordinates an equal number of times. This equal-coverage property is the key to proving that the failover routing scheme balances load evenly.

Below, we state the main theorem of this chapter.

Theorem 12. *Given any fixed throughput value $r \in (0, \frac{1}{2}]$, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$, and let*

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

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

Similar to Sections 6.1 and 6.2, we will begin by constructing an SORN design $\mathcal{S}^0$ which

is parameterized by N, g, and C, where C is a parameter which we set during our analysis to achieve the appropriate tradeoffs between throughput and latency. We will then analyze $\mathscr{S}_N(g, C)$, a distribution over all SORN designs $\boldsymbol{\mathcal{S}}^\tau$ which are equivalent to $\boldsymbol{\mathcal{S}}^0$ up to re-labeling of nodes, and show that it satisfies the conclusion of Theorem 12.

Before we define $\boldsymbol{\mathcal{S}}^0$, we first provide some intuition behind the design.

Definition 20. A (C, g)-*constellation* in $\mathbb{F}_p^g$ is a sequence of $C(g+1)$ vectors for which the following property holds. Any set of g distinct vectors forms a basis over the vector space $\mathbb{F}_p^g$.

The ORN design described in Chapter 6 was defined using phases of Vandermonde vectors. This was only done to achieve the property that any set of g vectors, each chosen from a different phase block, formed a basis over $\mathbb{F}_p^g$. No other special property of Vandermonde vectors was required. Thus, using any (C, g)-constellation gives the same throughput-latency tradeoffs found in Theorem 7.

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

Lemma 5. *Suppose $A \in \mathbb{F}_p^{g \times g}$ is an invertible matrix, and $\boldsymbol{\mathcal{V}} = (v_1, v_2, \ldots, v_{C(g+1)})$ is a (C, g)-constellation in $\mathbb{F}_p^g$. Then the sequence $A\boldsymbol{\mathcal{V}} = (Av_1, Av_2, \ldots, Av_{C(g+1)})$ is also a (C, g)-constellation in $\mathbb{F}_p^g$.*

Proof. Suppose not, that there exists some set of vectors $w_{i_1}, \ldots w_{i_g}$ each from different blocks of $A\boldsymbol{\mathcal{V}}$ which are linearly dependent. Then WLOG there exists constants $\alpha_1, \ldots, \alpha_{g-1}$ such that $\alpha_1 w_{i_1} + \ldots + \alpha_{g-1} w_{i_{g-1}} = w_{i_g}$. Then $\alpha_1 A v_{i_1} + \ldots + \alpha_{g-1} A v_{i_{g-1}} = A v_{i_g}$ for vectors

$v_{i_1}, \ldots, v_{i_g}$ each from different blocks of $\mathcal{V}$. This is a contradiction due to distributivity of matrix and vector multiplication, and because A is invertible. $\square$

7.1 Connection Schedule

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

Then by Lemma 5, $A\mathcal{V}$ is a (C, g)-constellation for any matrix $A \in \mathcal{A}$. Order the set of matrices $\mathcal{A}$ arbitrarily, so that $\mathcal{A} = \{A_0, A_1, \ldots, A_{(p-1)^{g-1}-1}\}$. We rotate through the (C, g)-constellations formed by matrices in $\mathcal{A}$ to achieve our connection schedule.

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

As described above, $\mathscr{S}_N(g, C)$ is a distribution over all SORN designs $\mathcal{S}^\tau$ which are

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

7.2 Routing Protocol

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

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

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

$(g+1)$-hop paths. In this case, we use the same distribution over routing paths as in Section 6.2, when considering the set of $C(g+1)$ phases all belonging to the (C, g)-constellation beginning after time t. Due to the added delay, paths of this type have maximum latency $2C(g+1)N^{1/g}$, exceeding the maximum latency cited in Section 6.2 by a factor less than 2.

2-hop paths. To describe the distribution over 2-hop paths, first consider the following. Given a fixed Vandermonde vector $\boldsymbol{v} \in \mathcal{V}$, consider the set of edges formed by $a \to a + sA\boldsymbol{v} = b$

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

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

Note that paths of this type always take both hops during consecutive distinct periods, or iterations, of the schedule. Thus, paths of this type will have maximum latency

$$2(p-1)^g C(g+1) + C(g+1)(p-1) \leq C(g+1)N^{1/h} + 2C(g+1)N \leq \tilde{\boldsymbol{O}}(N),$$

because $C = \tilde{\boldsymbol{O}}(\log N)$.

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

Note that S^0_σ must make one choice for all timesteps t: to either route on $(g + 1)$-hop paths or 2-hop paths. In Section 7.5, we discuss how to analyze a design which allows S^0_σ to

route flow on a combination of $(g+1)$-hop and 2-hop paths, depending on starting timestep t.

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

7.3 Latency-Throughput Tradeoff

Theorem 13. *Given a fixed throughput value r, let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$, and assume $\frac{1}{r} \notin \mathbb{Z}$. Let N be a prime power p^g for primes p exceeding $\max\left\{C(g+1), 2 + \frac{2}{1-\varepsilon}, \frac{g+3}{\varepsilon} - 2, \frac{2-\delta}{1-\delta}\right\}$, where $\delta = \frac{(g+1)^{1/g}}{(g+2-\varepsilon)^{1/g}}$. Also let $\gamma = \ln\left(\frac{g+2-\varepsilon}{g+1}\right)$, let $C = \frac{\log\log N}{\gamma^2}\ln(N)$, and let*

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

Then:

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

Note that if $\frac{1}{r} \in \mathbb{Z}$, then $\varepsilon = 1$, and there do not exist primes p for which $p \geq 2 + \frac{2}{1-\varepsilon}$. Thus, as in Chapters 4 and 6, we condition against $\frac{1}{r} \in \mathbb{Z}$.

Both parts of this theorem will be proven by focusing on the probability that S^0_σ must deviate from sending all flow on $(g+1)$-hop paths to sending all flow on 2-hop paths. This is

directly correlated with when congestion occurs on physical edges in the design $\mathcal{S}^0$, if we were to always send flow on $(g+1)$-hop paths. We note the similarities between $\mathcal{S}^0$ and $\mathcal{R}^0$ from Chapter 6, and apply the same exponential tail bounds of bilinear sums to get our result.

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

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

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

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

We first consider the failure probability of edges within each (C, g)-constellation individually. Fix an edge e and $0 \leq q \leq g$, and consider the amount of flow traversing edge e traveling on paths where edge e occurs in the $(q+1)$-th phase block of the flow path.

Note that, unlike in the proof of Theorem 8, edges e that appear in the $(q+1)$th phase block of a (C, g)-constellation, for $0 \leq q \leq g$, will *only* have $(q+1)$-th hop flow traversing e, due to delaying flow before routing by whole (C, g)-constellations instead of single phase blocks. Then the total amount of $(q+1)$-th hop flow traversing edge e equals the total amount of any-hop flow traversing edge e.

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

Now suppose $1 \leq q \leq g-1$, and let Y_i be the random variable realizing the amount of $(q+1)$-th hop flow traversing edge e due to source node i, normalized by $\frac{1}{g+1}$. Clearly, the total amount of $(q+1)$-th hop flow traversing e will be $(g+1)\sum_i Y_i$. The variables Y_i act exactly as the random variables X_i in Section 6.3, in the proof of Theorem 8. Therefore, the same tail bound conclusions about their sum are applicable.

Therefore, over the uniform distribution for the fixed design $\mathcal{S}^0$, and for the family of

distributions $\mathscr{S}_N(g, C)$, we have

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

Finally, we need to show that if none of the bad events as described above occur, if every edge has at most $e^{\gamma}r$ $(q+1)$-th hop flow for $1 \leq q \leq g-1$, then no edge will be overloaded.

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

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

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

□

7.4 Provably Separating ORN and SORN Capabilities

In this section we show that semi-oblivious routing has a provable asymptotic advantage over oblivious routing in reconfigurable networks. In order to do so, we must compare the guaranteed throughput versus latency tradeoffs achieved by the family of SORN designs

$\mathscr{S}_N(g, C)$ described above and distributions over ORN designs. We will show below that our family of SORN designs $\mathscr{S}_N(g, C)$ has a provable asymptotic advantage over ORNs in *average latency*. To do so, we provide the following lower bound on average (expected) latency of distributions over ORN designs.

Theorem 14. *Consider any constant $r \in (0, \frac{1}{2}]$. Let $h = h(r) = \lfloor \frac{1}{2r} \rfloor$ and $\varepsilon_o = \varepsilon_o(r) = h + 1 - \frac{1}{2r}$, and let $L_{obl}(r, N)$ be the function*

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

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

The proof of Theorem 14 follows a similar structure as the lower bound proof of Section 5.1.1, only with an added average latency constraint in the starting linear program, which results in an additional variable in the corresponding dual program, which must be reasoned about and assigned a value. We leave the proof to Section 5.3.

Theorem 15. *Consider any constant $r \in (0, \frac{1}{2}]$, and let $g = g(r) = \lfloor \frac{1}{r} - 1 \rfloor$ and $\varepsilon = \varepsilon(r) = g + 1 - (\frac{1}{r} - 1)$ Then if $r \in \left(0, \frac{1}{4}\right] \cup \left[\frac{1}{4-(2/N^{1/6})}, \frac{1}{3}\right]$ and $\frac{1}{r}$ is not an integer, the family of SORN designs $\mathscr{S}_N(g, C)$ achieves asymptotically better average latency than any family of ORN designs which guarantees throughput r.*

Proof. By Theorem 14, any family of ORN designs which guarantees throughput r must suffer average latency $\Omega\left(L_{obl}(r, N)\right)$. Also recall that the family of SORN designs $\mathscr{S}_N(g, C)$ achieves maximum latency $\tilde{\boldsymbol{O}}(gN^{1/g})$ with high probability as long as $\frac{1}{r}$ is not an integer. This implies it also achieves average latency $\tilde{\boldsymbol{O}}(gN^{1/g})$, since with probability 1 it achieves maximum latency $\tilde{\boldsymbol{O}}(N)$. We divide the set of throughput values $r \in \left(0, \frac{1}{4}\right] \cup \left[\frac{1}{4-(2/N^{1/6})}, \frac{1}{3}\right]$ into the following cases.

1. $r \leq \frac{1}{5}$. Then $g(r) > h(r) + 1$. Since $L_{obl}(r, N) \geq N^{1/(h+1)}$, then $L_{obl}(r, N)$ is asymptotically greater than $\tilde{O}(L^*_{upp})$.

2. $r \in \left(\frac{1}{5}, \frac{1}{4}\right]$. Then $\varepsilon_o(r) = h + 1 - \frac{1}{2r} \geq \frac{1}{2}$, and $g(r) > h(r)$. Therefore, $L_{obl}(r, N)$ is asymptotically greater than $\tilde{O}(L^*_{upp})$.

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

□

7.5 Mixing $(g+1)$-hop and 2-hop paths in our SORN Design

In defining the routing protocol of our SORN design in Section 7.2, we always chose to route permutation demand D_σ on 2-hop paths if *any* edge e in any (C, g)-constellation would become overloaded from routing D_σ on $(g+1)$-hop paths. However, this choice is a bit extreme. After all, the connection schedule iterates through $(p-1)^{g-1}$ different (C, g)-constellations.

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

This strategy slightly decreases the achievable throughput rate, due to reserving a small amount of edge capacity on each edge for 2-hop paths. However, as long as the number of contentious (C, g)-constellations k is small, we can still provably achieve throughput r for

any $r \in (0, \frac{1}{2})$ for which $\varepsilon(r) = \left\lfloor \frac{1}{r} - 1 \right\rfloor + 1 - (\frac{1}{r} - 1) \neq 1$. (Or in other words, for r which is not the reciprocal of an integer.)

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

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

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

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

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

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

For each edge $e = (a, b)$, consider the total amount of first-hop flow from 2-hop paths traversing the edge. First-hop flow traversing e must be traveling from source node $i = a$. Also note that since edge e exists in the network, then the vector $b - a$ must have only non-zero coordinates. Then first-hop flow traverses edge e only when $\sigma(a) - b$ also has only non-zero coordinates.

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

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

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

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

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

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

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

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

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

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

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

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

Now consider the probability that k (C, g)-constellations are contentious. This is clearly no more than the probability that a single (C, g)-constellation is contentious, which occurs with negligible probability as stated in the proof of Theorem 13.

□

BIBLIOGRAPHY

[1] 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.

[2] R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *IEEE Transactions on Information Theory*, 48(3):569–579, 2002.

[3] Slavisa Aleksic. Electrical power consumption of large electronic and optical switching fabrics. pages 95 – 96, 02 2010.

[4] Daniel Amir, Tegan Wilson, Vishal Shrivastav, Hakim Weatherspoon, and Robert Kleinberg. Poster: Scalability and congestion control in oblivious reconfigurable networks. ACM SIGCOMM '23, page 1138–1140, New York, NY, USA, 2023. Association for Computing Machinery.

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

[6] 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.

[7] 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.

[8] Roger C Baker, Glyn Harman, and János Pintz. The difference between consecutive primes, ii. *Proceedings of the London Mathematical Society*, 83(3):532–562, 2001.

[9] 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.

[10] Margaret Barthel. Northern Virginia's data center industry is booming. but is it sustainable?, 2023.

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

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

[13] Q. Cheng, A. Wonfor, J. L. Wei, R. V. Penty, and I. H. White. Demonstration of the feasibility of large-port-count optical switching using a hybrid mzi-soa switch module in a recirculating loop. *Opt. Lett.*, 39(18):5244–5247, Sep 2014.

[14] Jeff Clabaugh. Northern Virginia is again the no. 1 data center market, but challenges are mounting, 2024.

[15] Paolo Costa, Hitesh Ballani, Kaveh Razavi, and Ian Kash. R2c2: A network stack for rack-scale computers. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 551–564, New York, NY, USA, 2015. ACM.

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

[17] Michael Dinitz and Benjamin Moseley. Scheduling for weighted flow and completion times in reconfigurable networks. *CoRR*, abs/2001.07784, 2020.

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

[19] Dominion Energy. Q4 2022 earnings call, 2022.

[20] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, page 339–350, New York, NY, USA, 2010. Association for Computing Machinery.

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

[22] 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.

[23] Synergy Research Group. Hyperscale data center capacity to almost triple in next six years, driven by AI, 2023.

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

[25] 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.

[26] Su Jia, Xin Jin, Golnaz Ghasemiesfeh, Jiaxin Ding, and Jie Gao. Competitive analysis for online scheduling in software-defined optical WAN. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, 2017.

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

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

[29] 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.

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

[31] Jongman Kim, Chrysostomos Nicopoulos, Dongkook Park, Reetuparna Das, Yuan Xie, Vijaykrishnan Narayanan, Mazin S. Yousif, and Chita R. Das. A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In *Proceedings of*

the 34th Annual International Symposium on Computer Architecture, ISCA '07, page 138–149, New York, NY, USA, 2007. Association for Computing Machinery.

[32] 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.

[33] Zeph Landau and Alexander Russell. Random Cayley graphs are expanders: A simple proof of the Alon–Roichman Theorem. *Electr. J. Comb.*, 11, 09 2004.

[34] Sergey Legtchenko, Nicholas Chen, Daniel Cletheroe, Antony Rowstron, Hugh Williams, and Xiaohan Zhao. Xfabric: A reconfigurable in-rack network for rack-scale computers. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 15–29, Santa Clara, CA, 2016. USENIX Association.

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

[36] Hong Liu, Ryohei Urata, Kevin Yasumura, Xiang Zhou, Roy Bannon, Jill Berger, Pedram Dashti, Norm Jouppi, Cedric Lam, Sheng Li, et al. Lightwave fabrics: At-scale optical circuit switching for datacenter and machine learning systems. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 499–515, 2023.

[37] Macom M21605 Crosspoint Switch. `https://www.macom.com/products/product-detail/M21605/`.

[38] William M. Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C. Snoeren, and George Porter. Expanding across time to deliver bandwidth efficiency and low latency.

In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 1–18, Santa Clara, CA, February 2020. USENIX Association.

[39] Rich Miller. Dominion: Virginia's data center cluster could double in size, 2023.

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

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

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

[43] 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.

[44] Ankit Singla, Atul Singh, and Yan Chen. OSA: An optical switching architecture for data center networks with unprecedented flexibility. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 239–252, San Jose, CA, 2012. USENIX.

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

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

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

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

[49] Meg Walraed-Sullivan, Jitendra Padhye, and David A. Maltz. Theia: Simple and cheap networking for ultra-dense data centers. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, pages 26:1–26:7, New York, NY, USA, 2014. ACM.

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

[51] 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.

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