# Lanternfish: Better Random Networks Through Optics

Tyler Szepesi
University of Waterloo
stszepesi@cs.uwaterloo.ca

Bernard Wong
University of Waterloo
bernard@cs.uwaterloo.ca

Fiodar Kazhami
University of Waterloo
fkazhami@cs.uwaterloo.ca

Sajjad Rizvi
University of Waterloo
sm3rizvi@cs.uwaterloo.ca

Tim Brecht
University of Waterloo
brecht@cs.uwaterloo.ca

## ABSTRACT

Current random datacenter network designs, such as Jellyfish, directly wire top-of-rack switches together randomly, which is difficult even when using best-practice cable management techniques. Moreover, these are static designs that cannot adapt to changing workloads. In this paper, we introduce Lanternfish, a new approach to building random datacenter networks using an optical ring that significantly reduces wiring complexity and provides the opportunity for reconfigurability. Unlike previous optical ring designs, Lanternfish does not require wavelength planning because it is specifically designed to provide random connectivity between switches. This design choice further reduces the difficulty of deploying a Lanternfish network, making random datacenter networks more practical. Our experimental results using both simulations and network emulation show that Lanternfish can effectively provide the same network properties as Jellyfish. Additionally, we demonstrate that by replacing passive optical components with active optical components at each switch, we can dynamically reconfigure the network topology to better suit the workload while remaining cost competitive. Lanternfish is able to construct workload specific topologies that provide as much as half the average pathlength than a Jellyfish deployment with twice as many switch-to-switch connections.

## 1. INTRODUCTION

Random datacenter networks are scalable and fault-tolerant with small average network diameters and high bisection bandwidth [1, 2]. In theory, they are an excellent alternative to existing tree-based network topologies, yet they are not used in practice outside of small research deployments. One of the main barriers to adoption is the difficulty in wiring and administering a datacenter-scale random network.

The current approaches to building a random datacenter network are inordinately arduous. For example, building a Jellyfish [1] network requires connecting each top-of-rack (ToR) switch to a constant number of other randomly selected ToR switches that are likely dispersed throughout the datacenter. Not only is this taxing to physically deploy, it also makes simple cable management techniques, such as cable bundling and horizontal cable managers, largely ineffective because nearby ToR switches are not connected to the same remote switches. Extensive use of patch panels is therefore necessary to avoid creating a datacenter-wide spiderweb of cables. However, even with a patch panel for each row of racks, wiring the random connections within a row and between patch panels across rows can be exacting.

In this paper, we introduce Lanternfish, a new approach to building random datacenter networks that leverages commodity photonics to significantly reduce wiring complexity and introduce a degree of reconfigurability. Similar to past work on implementing a full mesh network using an optical ring [3, 4, 5], Lanternfish assigns a fixed number of wavelengths to each ToR switch, and uses optical multiplexers and demultiplexers to connect the switches to an optical ring. The multiplexers combine wavelengths to allow a single cable to carry multiple signals, and the demultiplexers extract wavelengths that are appropriate for the switch at each hop in the ring. However, unlike full mesh networks that require judicious wavelength planning to create the full mesh structure, Lanternfish can instead perform random wavelength selection at each switch because it is specifically trying to provide random connectivity between switches. Wiring a Lanternfish network is therefore far simpler compared to wiring other random datacenter networks.

The key challenge in building a Jellyfish-like network using Lanternfish is in providing uniform random connectivity between switches. Intuitively, since two switches $a$ and $b$ are only connected using wavelength

Technical Report CS-2015-11
David R. Chariton School of Computer Science
University of Waterloo, Canada
.

$\lambda$ if they have exclusive use of $\lambda$ within at least one of the ring segments between $a$ and $b$, the probability that two switches are connected is therefore a function of the spectrum size and the number of optical hops between the switches. We can ensure that Lanternfish provides uniform random connectivity by using a separate wavelength for each connection. However, this is impractical since an optical ring can carry at most 160 wavelengths using 50 GHz channel spacing [6]. We show that significantly fewer wavelengths are required to have near-uniform random connectivity and provide network performance characteristics that are effectively equivalent to that of Jellyfish.

For large datacenters with thousands of ToR switches where the available wavelengths from a single optical ring are insufficient, we can naïvely increase the effective number of available wavelengths by connecting the switches to multiple optical rings. However, this solution is not cost-effective when many rings are necessary due to the cost of requiring additional multiplexers for each ring. Instead, Lanternfish only connects each switch to a small randomly selected subset of the rings. This approach significantly reduces the required number of multiplexers while still providing near-uniform random connectivity.

Although random networks provide excellent network properties for arbitrary endpoints, most datacenter workloads exhibit significant network locality [7] and would benefit from having better network connectivity between nearby racks in exchange for a small reduction in cross-datacenter network performance. Furthermore, a static random network may not be well-suited for latency-sensitive applications that require a small average or maximum hop-count between a specific host set. Biasing the randomness of a Jellyfish network to cater to a particular workload or host set would be disastrous if the workload or hosts ever change, since it would either lead to poor network performance or require rewiring large portions of the datacenter network.

To address the problem of changing network requirements, we explore the use of commodity active optical devices, namely wavelength selective switches, to equip Lanternfish with the ability to incorporate topological reconfiguration. By replacing optical multiplexers with wavelength selective switches, we can programmatically change which ring a transceiver is connected to without any physical rewiring. This degree of flexibility is sufficient to significantly alter the network topology and provide performance benefits.

We evaluate Lanternfish through both simulations and network emulation using Mininet [8]. Our results demonstrate that Lanternfish can provide network properties that are effectively equivalent to that of Jellyfish while requiring only a small number of optical rings. Moreover, we show that Lanternfish can, for non-uniform workloads, provide significant workload specific advantages over Jellyfish by reconfiguring its topology. Our cost analysis finds that Lanternfish is already cost-competitive with Jellyfish for large networks. More importantly, given current cost-trends, reconfigurable Lanternfish will likely offer a cost advantage over static networks for many workloads in the near future.

Overall, our work makes three contributions:

- We present Lanternfish, a new optical approach to building random datacenter networks with insignificant wiring complexity.
- We show that Lanternfish can scale to large networks, and provide reconfigurability using commodity optical devices while being cost-competitive.
- We evaluate the performance of Lanternfish and find that it performs as well as Jellyfish, and through topological reconfigurations can adapt to changing workload and outperform Jellyfish.

## 2. BACKGROUND AND RELATED WORK

In this section, we survey previous work on datacenter network topologies, random networks, and optical networks. We also describe the commodity optical devices that Lanternfish uses to implement a random network in an optical ring.

### 2.1 Network Topologies

Traditional datacenter networks are constructed as a multi-rooted tree. The basic design is to have three layers of switches, with the lowest layer connected directly to the servers in a rack (edge), middle layer switches aggregating the ToR switches (aggregate), and the top (core) layer providing cross datacenter connectivity. There are two common problems with multi-rooted tree topologies: over-subscription, and long path lengths. Over-subscription is the result of an imbalance in network capacity where, for a given layer, its total link capacity to the layer below it (e.g., aggregation to edge) is higher than its total link capacity to the layer above it (e.g., aggregation to core). Long path lengths occur when packets need to traverse multiple layers of the network to reach non-local destinations.

Given the limitation of multi-rooted tree topologies, there have been many alternative topology proposals, some of which include: BCube [9], DCell [10], fat-tree [11, 12], VL2 [13], and Jellyfish [1]. Despite the large number of alternative designs, most datacenter networks continue to be based on a multi-rooted tree topology. This is due to a combination of datacenters operators being conservative with their choice of network topologies, and perhaps more importantly, the significant deployment challenges for many of these topolo-

gies, such as wiring complexity, that outweigh their benefits.

## 2.2 Random Networks

In contrast to the highly structured approach taken by most new datacenter network topologies, Jellyfish [1] proposes randomly interconnecting ToR switches. This eliminates the need to carefully plan the connectivity between switches, while offering higher bisection bandwidth and smaller average network diameter than other network designs with the same amount of hardware. Furthermore, random networks have been shown to provide network bandwidth that, for a given number of internal switch-to-switch links, approaches the theoretical upper bound [14].

A major drawback of random networks is the nontrivial increase in deployment complexity due to the difficulty in wiring ToR switches together randomly. While structured networks may have performance limitations, many of them are far easier to build and maintain. An additional drawback to random networks is that they perform poorly compared to tree-based topologies for workloads that exhibit strong network locality. Lanternfish addresses this drawback by introducing reconfigurability, which enables the switch-to-switch connections to be tailored to a specific workload. In the case of workloads with strong network locality, Lanternfish would increase its ratio of nearby to distant connections.

## 2.3 Optical Networks

Current datacenter networks rely almost entirely on electrical networking technology. Although fiber optic connections are often used between switches to reduce transmission power and improve signal quality, the switches are electrical and no other optical components are used in the network.

One approach to leverage photonics in a datacenter network is to use optical switches that can provide a direct optical connection between any port pair. The most common technology for building optical switches is microelectromechanical systems (MEMS), which connects port pairs by mechanically aligning mirrors. Unfortunately, optical switches currently suffer from relatively high switching latencies [15, 16] and are limited in scale. An alternative approach is to build a *wavelength division multiplexing* (WDM) network. The defining characteristic of a WDM network is its use of multiple wavelengths on a single fiber optic cable, which allows multiple independent communication channels to share a single physical cable. This is possible through the use of optical multiplexers/demultiplexers (muxes/demuxes).

The simplest and least expensive type of optical mux/demux uses *Array Waveguide Grating* (AWG). An AWG connects a single port that carries all $W$ wavelengths to $W$ other ports, each of which carries one of the $W$ wavelengths. A *Wavelength Selective Switch* (WSS) is a functionally more complex optical mux/demux. Instead of having $W$ ports that each carry a single wavelength, there are $N$ ports (where $N \leq$ W) that each carry a subset of the $W$ wavelengths. Additionally, WSSes are reconfigurable, so it is possible to change which wavelength maps to which port programactically without having to modify the wiring.

Another useful device is an optical *splitter*, which forwards an incoming signal to multiple output ports. The power of the incoming signal is split evening across the output ports. In combination with a WSS, this allows a signal to be broadcast to many destinations, and then have individual wavelengths dynamically selected at each source.

A *transceiver* is used to bidirectionally convert between electrical and optical signal, with the optical signal being set to a particular wavelength. It provides connectivity between the optical and electrical networking components.

Previous work has explored how optics can be used in datacenter networks. Two datacenter network designs are Helios [17] and c-Through [18], both of which incorporate optical switching into a traditional multi-rooted tree topology. c-Through does this by connecting ToR switches to a single optical switch, while Helios replaces a subset of core switches with optical switches. Although they are interesting points in the design space, these networks are still fundamentally based on tree topologies and only provide performance benefits to flows that are selected to use the optical part of the network. Because of the relatively long switching delays of optical switches, only long "elephant" flows can take advantage of these designs. Furthermore, their reliance on a set of large centralized optical switches can limit their scalability. Most commercial optical switches are currently limited to a few hundred ports [19].

An alternative approach is OSA [19], which uses both WDM and optical switches to completely replace electrical switches. OSA connects all ToR switches to a single optical switch. However, unlike c-Through, there are multiple connections between the optical switch and each ToR switch. Additionally, each ToR output port can be reconfigured to use a different connection to the optical switch by using a WSS. This is a very flexible design, but it has severe scalability limitations since it requires multiple connections between each ToR switch and a single optical switch.

As with OSA, Mordia [20] is a fully optical network design that connects all ToR switches to a single optical switching component. Unlike OSA, Mordia uses a ring of WSSes to provide the equivalent of a single optical switch, for the purpose of reduced switching la-

tency, and each ToR switch has a single connection to the optical ring. Connectivity between switches is provided by circuit switching an individual wavelength for each switch. While providing a different form of flexibility than OSA, Mordia is still limited to medium sized networks.

Finally, Quartz [3] presents another fully optical network design using only WDM muxes/demuxes. It creates a full mesh network by forming a ring with the ToR switches, and requires careful wavelength assignment to each switch in order to ensure that there is a direct optical connection between every switch pair. The focus of this design is to eliminate cross-traffic congestion and reduce switching delay by providing direct connectivity. However, its full mesh requirement limits its scale to only hundreds of switches. Its intended use is to replace parts of a large datacenter network, rather than the entire network.

## 3. RANDOM OPTICAL NETWORKS

In designing Lanternfish, we strive to maintain the performance characteristics of a random network like Jellyfish, while simplifying the physical deployment and maintenance complexity. Previous work on random datacenter networks [1] [2] describe the structure of the network using connections that are created by a physical wire between two endpoints. In this section, we describe how our optical ring design can be used to create the equivalent random structure and show, using statistical analysis, that it can provide topological characteristics similar to Jellyfish.

### 3.1 Lanternfish Design

A Lanternfish network is a collection of Lanternfish switches organized into a ring where adjacent switches in the ring are connected by an optical cable. Each Lanternfish switch consists of a ToR switch, a set of transceivers with randomly selected wavelengths, and a pair of AWG multiplexer/demultiplexers. This base version of Lanternfish is very similar in design to that of a Quartz network [3]; the two network designs primarily differ in the way they select transceiver wavelengths. We will see in Section 3.4 that, in order for this design to scale and workaround wavelength limitations, we need to introduce new design elements that are unique to Lanternfish.

Figure 1 shows an example of two adjacent switches in a ring with 4 wavelengths. The diagram shows the optical components for a single direction, and the multiplexers are duplicated to provide communication in the opposite direction around the ring. Each switch has a total of 10 ports, with 6 ports (shown at the bottom of each switch) connected to servers in the rack and 4 ports (shown at the top of each switch) connected through transceivers to the optical ring. The AWGs demultiplex incoming signals and forward a wavelength to the switch if there is a transceiver using a matching wavelength. The remaining signals are forwarded to the opposing AWG; they pass directly through this Lanternfish switch to the next switch in the ring. The Lanternfish switch sends outgoing signals from the transceivers to an AWG, which multiplexes them together with the pass-through signals before sending them to the next switch. To ensure that the AWGs cannot receive two signals using the same wavelength, the transceivers are organized as pairs with each pair using the same wavelength, and each transceiver in the pair is connected to an opposing AWG. Therefore, a Lanternfish switch cannot both pass-through a signal with wavelength $\lambda$ and generate an outgoing signal using wavelength $\lambda$.

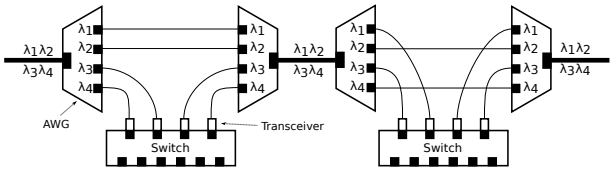


**Figure 1: Neighbouring switches, 4 wavelengths and 4 transceivers per switch.**

The randomness in Lanternfish comes from the selection of wavelengths for each pair of transceivers. For example, if there are $T$ transceivers at a switch, and $W$ wavelengths to choose from, then $T/2$ unique wavelengths are selected (one for each pair of transceivers), with equal probability. The remaining $W - T/2$ wavelengths, which are not assigned to transceivers, are referred to as pass-through wavelengths. These wavelengths are not destined for this switch and go directly from one AWG to the other.

Optical connections between switches are formed by sharing a wavelength in the optical ring. For example, if a transceiver uses wavelength $\lambda_1$ to connect to the optical ring, then the first switch in the clockwise as well as the counter-clockwise direction using wavelength $\lambda_1$ will be optically connected to this switch. Therefore, by randomly assigning the wavelengths used by each switch, we are indirectly randomizing the connectivity with other switches.

### 3.2 Lanternfish Properties

In a Jellyfish topology, there are three parameters to consider: the number of switches $S$, the number of ports per switch $P$, and the number of ports connected to switches $T$. Together, these parameters determine how many servers are supported by the topology, which is $S(P - T)$. It also determines a network's level of connectivity (which is primarily a function of $T$). For a Lanternfish topology, we also consider the number of wavelengths in the optical ring.

To understand the impact of $W$ on the connectivity

properties of Lanternfish, recall that there is an optical connection between two switches only if they support a wavelength $\lambda$ that is not supported by other switches between them in the clockwise or counter-clockwise direction. Therefore, for each wavelength that a switch supports, it is connected to the *first* switch along the ring that shares the wavelength. This implies that there is a higher probability of forming connections to closer switches. The amount of bias in the probability distribution towards local connectivity is a function of $W$. Increasing the number of wavelengths reduces the number of switches that share a wavelength, which increases the average distance between optically connected switches. This differs from Jellyfish, which has uniformly random connectivity across switches in the network.

To evaluate the effect of $W$ on the network, we generate 100 random graphs and measure and report the averages of the mean and maximum path lengths. We vary the number of wavelengths and compare these metrics with 100 randomly generated Jellyfish topologies. Together, these represent the common and worst case scenarios for the connectivity between two switches.

Figure 2 shows the averages of the mean and maximum path lengths for topologies that contain 128 switches, each with 6 transceivers per switch and a range of wavelengths in the optical ring. With 20 wavelengths, both the mean and maximum path lengths are longer for Lanternfish than Jellyfish, which is an artifact of heavily favouring connections between switches in close proximity to each other along the ring. However, Lanternfish networks using between 40 to 80 wavelengths provide effectively the same path lengths as Jellyfish.

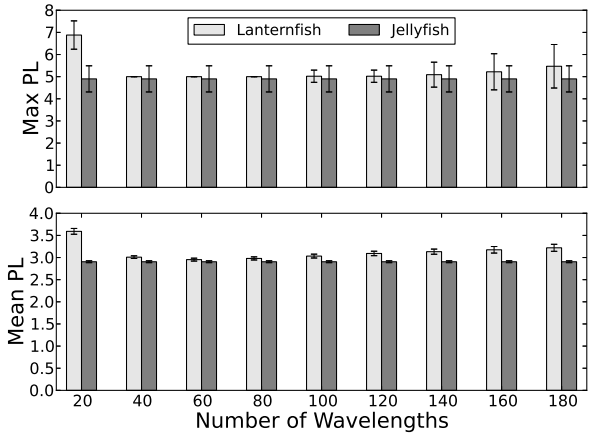


**Figure 2: Path lengths for 128 switch topologies, 6 transceivers per switch.**

For Lanternfish, although the differences are not statistically significant, the average maximum path length appears to increase slightly when more than 80 wavelengths are used. This represents the other extreme for Lanternfish, where a faction of the wavelengths are used by only two switches. The problem here is that, since a Lanternfish switch mirrors the transceiver wavelengths to support the same wavelengths in both directions, duplicate connections between switches are formed if only two switches in the entire ring share a wavelength (one in either direction around the ring). This reduces path diversity, which in turn increases the maximum network path length.

Figure 3 compares different path lengths using larger topologies, with 1024 switches and 12 transceivers per switch. Of note is the fact that larger topologies have a proportionally wider range of wavelengths (800-1400) that essentially provide the same network characteristics as Jellyfish.

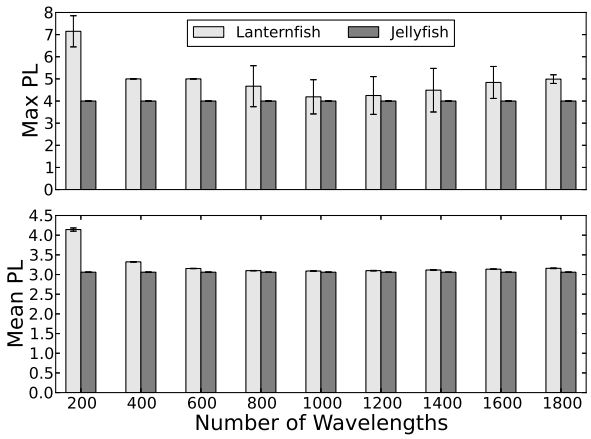


**Figure 3: Path lengths for 1024 switch topologies, 12 transceivers per switch.**

## 3.3 Loss In Signal Strength

As pointed out in previous work [3], one of the subtle challenges of building a WDM network is that optical signals degrade due to insertion loss when passing though a multiplexer. Therefore, signal amplifiers may be required to maintain the signal strength. The number of amplifiers needed throughout the ring is determined by the output power and receiver sensitivity of the transceivers, and by the signal loss of each multiplexer.

Consider an example where transceivers output a signal at 4 dBm and have receiver sensitivity of -15 dBm [21], and multiplexers have a 4.2 dB insertion loss [22]. A signal can pass through 4.5 multiplexers $(((4\text{ dBm} - (-15\text{ dBm}))/4.2)$ dB) without requiring any amplification. Additionally, inexpensive attenuators may be necessary to avoid overloading transceivers and amplifiers.

## 3.4 Adding Rings

There are two important problems with the basic de-

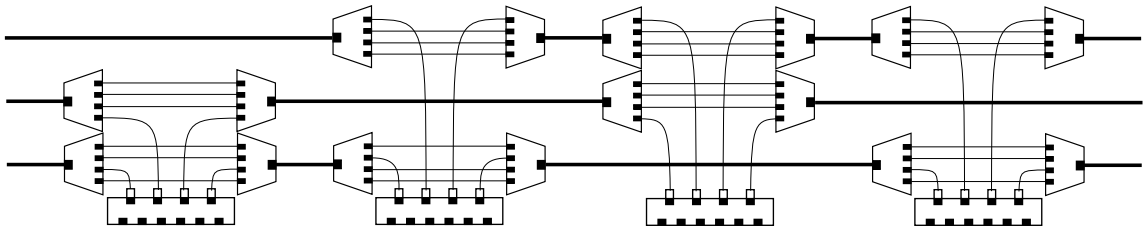

**Figure 4: Four neighbouring switches, using 3 rings, 4 wavelengths per ring, and 4 transceivers per switch.**

sign of Lanternfish. First, a single AWG failure will convert the ring into a line, which greatly reduces the connectivity of the network. Second, for large networks, in order to provide near uniform random connectivity, Lanternfish requires more wavelengths than a single fiber cable can currently carry [6]. Although these are distinct problems, they can both be solved by adding more optical rings to the network.

Lanternfish can connect each switch to an additional optical ring by adding a pair of AWGs per switch. To ensure that Lanternfish maintains near uniform random connectivity between switches with multiple rings, we randomly select both a wavelength and a ring for each transceiver pair. For simplicity, we do not allow two pairs of transceivers in a switch to use the same wavelength on different rings.

Figure 4 gives an example of four neighbouring switches in a topology that consists of three rings, where each switch randomly connects to two of the three rings. As in Figure 1, each switch has 4 transceivers, and each AWG supports 4 wavelengths. Each ring is effectively providing additional wavelength choices for the network because the same wavelength is repeated on multiple rings.

Of potential concern is the number of AWGs that are introduced by adding more rings, because a pair of AWGs must be added for each ring that a switch connects to. Continuing with the previous example, if 24 transceivers are used on a switch, then it is possible for the switch to have connections to all 12 rings, which in turn requires 24 AWGs at the switch. As we will show in Section 5, placing a limit on the total number of AWGs in the network is both possible and has minor impact on performance.

Adding optical rings not only increases scalability and reduces the impact of link failures, it also improves the resilience of Lanternfish to AWG failures. In a Lanternfish network with a single ring, a faulty AWG that multiplexes $W$ signals will disconnect up to $W$ optical connections out of $(S \times T)/2$ total connections where $S$ is the total number of switches in the network and $T$ is the number of transceivers per switch. However, in a network with $k$ rings where each AWG only multiplexes $W/k$ wavelengths, the number of optical connections disconnected by a faulty AWG will be reduced by a factor of $k$.

## 4. DYNAMIC LANTERNFISH

Although passive AWGs allow us to build random optical networks that mimic the characteristics of a conventional random network, such as Jellyfish, it is also possible to use active WSSes in place of AWGs for multiplexing/demultiplexing. There are two major advantages to using a WSS instead of an AWG: demultiplexed wavelengths can share a port (instead of each wavelength having its own port) and a WSS can dynamically reconfigure wavelength assignments to output ports, without having to physically change the wiring.

### 4.1 Design

By incorporating WSSes into the design of Lanternfish we can providing a degree of topological reconfigurablity. Figure 5 depicts an example of a switch in a two ring Lanternfish topology that uses WSSes. In this example there are two optical rings, and the switch has 4 ports connected into the optical ring through transceivers. Unlike the previous diagrams, Figure 5 shows the full bidirectional design, as the incoming and outgoing signals use different components.

The first modification to the static design (as depicted in Figure 4) is in how the optical signals from the rings are handled. The AWGs that were attached to the optical rings in the outgoing direction have been replaced with WSSes. By using the ability to multiplex multiple wavelengths onto the same output port, each of these WSSes only needs two output ports: one for the pass-through wavelengths, and the other for the wavelengths used by the switch. In the incoming direction of the optical rings, the AWGs have been replaced with optical splitters that transmit all of the signal for all wavelengths as if they are both pass-through and being used by the transceivers. The wavelengths are then filtered by the WSS before being forwarded to the neighbouring switch or transceiver.

The second modification is the addition of components between the transceivers and the optical rings.

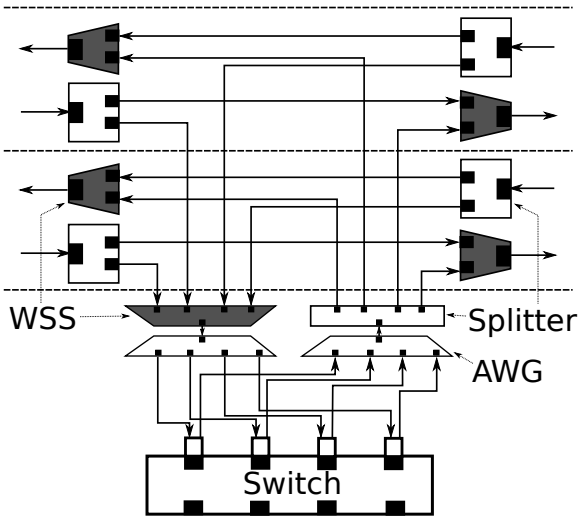


**Figure 5: A dynamic Lanternfish switch.**

The optical signal incoming from a ring first passes through a pair of multiplexers (one WSS and one AWG). The purpose of this pair of multiplexers is to filter incoming wavelengths, and then demultiplex the selected signals into individual wavelengths for the transceivers. Outgoing signals from the transceivers are first multiplexed together by an AWG and then broadcasted to all of the WSSes attached to the optical rings using an optical splitter. Notice that this only works if two transceivers cannot use the same wavelength. If two transceivers did use the same wavelength on different rings, the wavelengths would conflict when passing through the AWG.

## 5. EVALUATION

In this section, we evaluate the throughput and latency of static Lanternfish, as well as the reconfigurability of dynamic Lanternfish. Our objective is to demonstrate that a static Lanternfish deployment provides effectively the same performance properties as an equivalent Jellyfish network, and quantify the degree of reconfigurability in dynamic Lanternfish.

### 5.1 Throughput

We use two approaches to determine the throughput of Lanternfish and Jellyfish: simulation and emulation. The first evaluation is performed using the topobench flow simulator, which is designed to provide a fair and accurate comparison of the throughput of different network topologies [23] and which has been used by others to compare throughput [14]. Given the specification of a topology, topobench determines the largest capacity (with each link in the network offering a capacity of 1) that the minimum flow in the network is capable of receiving under a particular traffic matrix. In this paper, we give results for the random matching traffic matrix, where each server exchanges data with another random server in the network. We also ran simulations with all-to-all traffic and found similar relative results; the graphs for the second set of simulations are omitted due to space constraints.

Figure 6 shows the throughput for networks of different sizes, where each switch has 12 ports. 6 out of the 12 ports are connected to servers and the remaining 6 are connected to other switches (directly in the case of Jellyfish, and indirectly through the optical ring in the case of Lanternfish). From this graph, we see that 20 wavelengths provides throughput equal to that of Jellyfish in a network of 40 switches, but quickly falls off as the number of switches increases to 80. Alternatively, 60 wavelengths is not able to provide the same throughput as Jellyfish for smaller topologies, but in the range of 120 to 160 switches it performs just as well as Jellyfish. This matches our analysis in Figure 2, which shows that 60 wavelengths provides mean and max path lengths that are essentially the same as Jellyfish for topologies with 128 switches. Also, as we had expected, the relative throughput of Lanternfish to Jellyfish begins to fall off beyond 160 switches with 60 wavelengths, but the relative throughput actually improves with 120 wavelengths.

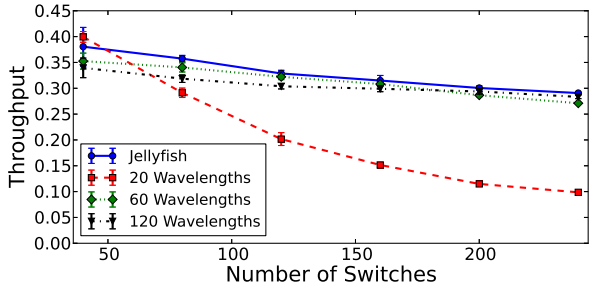


**Figure 6: Random Matching: 6 transceivers, 6 servers.**

To demonstrate how the use of multiple rings affects throughput, Figure 7 shows throughput results when 6 rings are used, with 10 wavelengths used in each ring, which is equivalent of 60 wavelengths on a single ring. Additionally, we place a cap on the total number of rings a particular switch can connect to in order to demonstrate the impact of restricting the total number of AWGs in the network (to reduce costs). Each line represents the maximum number of rings each switch is connected to, with Jellyfish shown as a baseline. The results show that throughput is only minimally affected by limiting the number of rings a switch can connect to, especially in larger datacenter networks with more than a hundred switches. Therefore, these results show that using multiple rings and limiting the number of ring connections per switch can improve Lanternfish scalability.

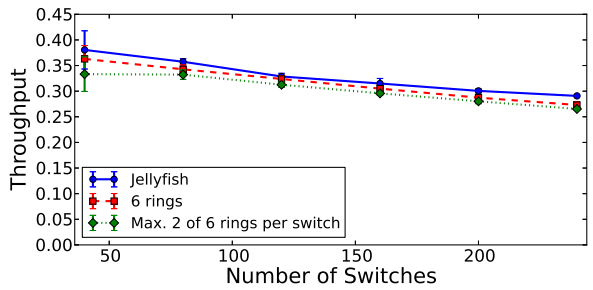


**Figure 7: Random Matching: 6 transceivers, 6 servers.**

Our second throughput evaluation is performed using Mininet [8] to emulate real traffic on Lanternfish topologies. The experiments were performed on an 8 core Intel(R) Xeon(R) E5-4640, with 16 GB of memory. In order to ensure that insufficent do not affect our measurements, our topologies are restricted to 10 Mbps links, 64 switches, 2 servers connected to each switch, and 4 transceivers per switch. Because our focus is on Lanternfish's relative performance to Jellyfish, the reduced link capacity should not affect our experimental conclusions. In these experiments, we apply a common scatter/gather traffic pattern to each network and use the workload completion time to evaluate the network's effective throughput. We vary the number of scatter/gather sources to find the network saturation point. Each scatter task sends 625 KB of data to each of 36 random destinations, and then retrieves 625 KB of data from each of these destinations as part of the gather task.

Figure 8 shows our performance results for 5, 15, 30, and 45 wavelength Lanternfish topologies. While having only 5 wavelengths is insufficient for this traffic matrix, the other topologies perform similarly to Jellyfish, with 20 wavelengths showing the lowest completion times among the various Lanternfish topologies.

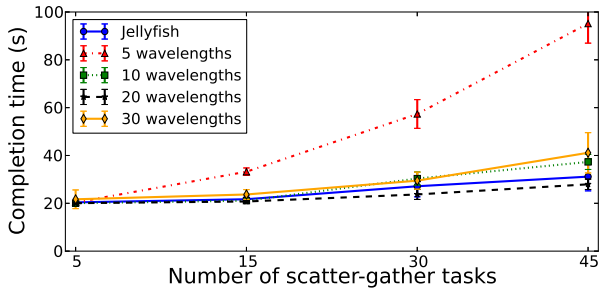


**Figure 8: Mean run times for different numbers of wavelengths (emulated).**

## 5.2 Latency

We perform our latency experiments using the packet-level discrete event simulator from Quartz [3]. Our graphs present the mean latency of a packet for a system in steady state. We omit the results where the network is fully saturated as queueing delays become unbounded when the system is no longer at steady state, and our focus for these experiments is in determining latency rather than throughput. All of the topologies use networks with 128 switches, 6 ports connected to other switches, and 6 ports connected to servers. All of the links in the network have a capacity of 10 Gbps and each switch has a switching latency of 380 nanoseconds. To generate load, we use scatter/gather traffic, with each source sending 400 byte packets at a rate of 150 Mbps to 60 destinations in the network. We demonstrate results for both global and local scatter/gather patterns. The global traffic pattern randomly selects destinations from all servers in the network, whereas the local traffic pattern selects the 60 closest servers to the source along the ring.

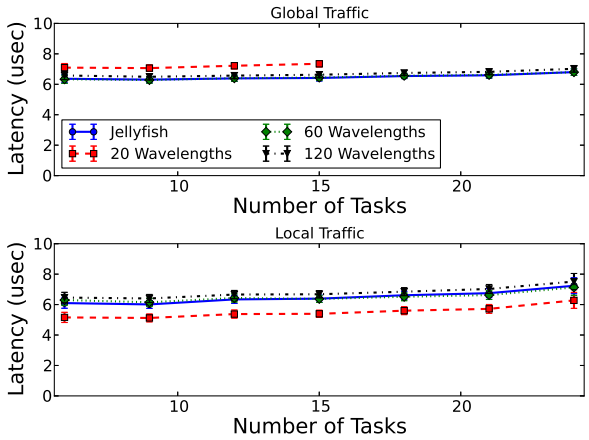


**Figure 9: Scatter/Gather: 128 switches, 6 transceivers per switch.**

The first set of results, shown in Figure 9, is a comparison of Lanternfish, with different numbers of wavelengths, to Jellyfish. The results show that a 60 wavelength Lanternfish topology produces latencies that are very similar to that of Jellyfish, which matches our path analysis results in Section 3.2. Interestingly, though it has poor global latency, a Lanternfish topology built with 20 wavelengths has lower mean latency for local traffic than any other configuration. With only 20 wavelengths, the probability of two switches that are in close proximity (within the range of the nearest 60 servers) being connected is very high. The result is very low path lengths between nearby servers, and thus lower latency than the other networks.

To determine the impact of using multiple rings on latency, we split 60 wavelengths across 6 rings with 10 wavelengths used in each ring. Figure 10 shows that the change in latency is statistically insignificant, even

when only of 2 of the 6 rings are attached to each switch.

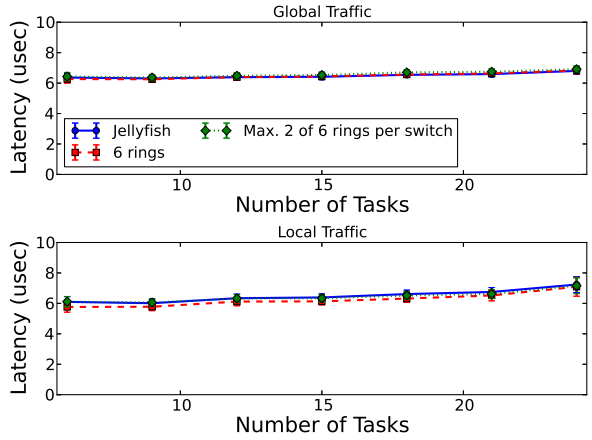


**Figure 10: Scatter/Gather: 128 switches, 6 transceivers per switch.**

Lastly, we examine the effectiveness of the dynamic Lanternfish design by reconfiguring it to increase the number of local connections. Figure 11 presents the latency results for the reconfigured Lanternfish topologies which can be compared with the results using the original Lanternfish topologies in Figure 10. By reconfiguring the topology, the latency drops for local traffic, regardless of the maximum number of rings attached to each switch. The cost of the local traffic latency reduction is a small increase in global traffic latency, which is expected when a topology is configured to favour local traffic.

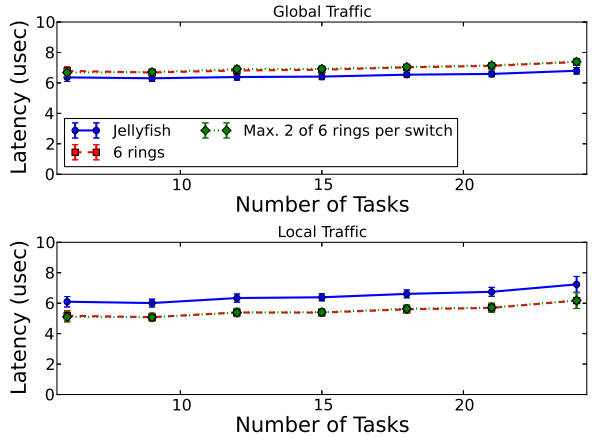


**Figure 11: Scatter/Gather: 128 switches, 6 transceivers per switch.**

We have demonstrated that properly designed Lanternfish networks are able to produce throughput and latencies that are on par with those of Jellyfish. A key contribution of our work is that this performance is obtained with significantly lower wiring complexity.

## 5.3 Reconfigurability

To evaluate the benefits of dynamic Lanternfish, we consider how well it is able to handle a range of workloads. Using its ability to reconfigure the connectivity between switches, dynamic Lanternfish is able to provide topological configurations specifically tailored to a given workload. In this way, pathlengths in a dynamic Lanternfish network are shorter than a Jellyfish network with double the number of switch-to-switch connections.

For this experiment, we specify our workload in terms of the subset of switches that each switch communicates with the most. For maximum performance, the switch would have direct (1 hop) connections with all other switches in its working set. Consequently, we represent the workload as a set of switch pairs, where each pair would have maximum performance with a 1 hop connection.

We evaluate how well Lanternfish is able to provide the required connectivity by giving it a *score* for each direct connection required. The score is equal to the number of additional hops between the two switches (the number of hops in the network, minus 1 for the required hop).

Unless otherwise stated, the following analysis is performed on a network with 512 switches. We construct 10 different randomly generated networks for each configuration, and subject each network to a set of 100 different demand graphs, with each switch requesting direct connectivity with 10 other switches. Each Lanternfish ring uses 88 wavelengths, as commodity WDM network devices are able to provide multiplexing of 88 wavelengths on an individual optical fiber, with each individual wavelength providing 10 Gbits of bandwidth [20].

The Lanternfish networks use 10 transceivers connected into the ring at every switch. The Jellyfish networks are built with double the number of transceivers at every switch. We provision each Jellyfish switch with more transceivers to compensate for its lack of reconfigurability.

We use a simple greedy heuristic solver to assign the transceivers at each switch to a particular ring in either the clockwise or counter-clockwise direction. The algorithm starts by attempting to create as many one hop connections as possible, based on the given connectivity demand. It starts by assigning transceivers to rings for the demanded connections between direct neighbours. Not all connections are possible, even between direct neighbours, because each switch has a statically assigned subset of the possible wavelengths. If two switches do not share a wavelength, there is no way for them to be directly connected. The algorithm then expands its assignment to connects switches of increasing distance along the ring, until all possible one hop connections have been assigned.

For the remaining transceivers that have not been assigned to a ring we simply form a connection with the closest switch along the optical ring that has an unused transceiver of a matching wavelength. A potential improvement to this algorithm would be to seek out two hop connections for the demanded connectivity.

### 5.3.1 Random Connectivity

We first consider the ability of Lanternfish to meet the workload demand over a range of rings, where every switch is connected to every ring. Figure 12 presents the CDF for the average number of extra hops that a demanded connection requires. With 3 optical rings, Lanternfish is not able to provide connectivity suited to the workload. Forming a connection between two switches on opposite sides of the optical ring is very expensive in term of wavelength utilization, as it eliminates the possibility of any other switches along the path using the same wavelength on the same ring.

With 5 or more rings, there is sufficient diversity of wavelengths and flexibility among rings to allow Lanternfish to adapt the topology and match a given workload better than a Jellyfish network with two times more switch-to-switch connections. Additionally, while adding more rings improves Lanternfish's ability to adapt to various workloads, there is a point of diminishing returns. In the extreme, additional rings provide no benefits because the only required one hop connections that are not satisfied are impossible due to the assignment of wavelengths to transceivers.

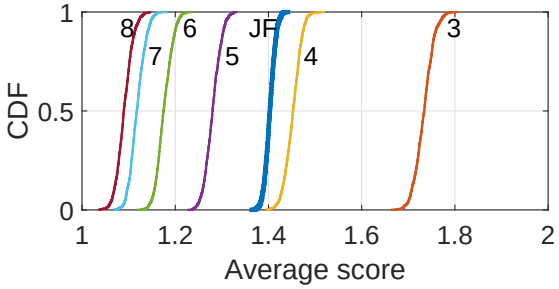


**Figure 12: Network score over 3 to 8 rings (labelled).**

While having 8 rings provides the shortest paths, it is also the most expensive network. WSSes contribute a large portion to the cost of a Lanternfish switch, with a switch requiring two WSSes for each ring that it connects to. In order to manage the cost of deploying a Lanternfish network, while still providing a large diversity of paths through the optical ring, the switches can be connected to a strict subset of the rings.

Figure 13 shows how the score of dynamic Lanternfish is affected when each switch is connected to a random subset of 8 total rings. Connecting each switch to 2 rings does not provide sufficient options for Lanternfish to reconfigure the connectivity. Each transceiver must select one of only two choices of rings, and the target for direct connectivity must not only use the same wavelength, but also share at least one of two rings to which the source is connected.

By connecting to only 4 of 8 rings, Lanternfish is able to produce shorter path lengths for a given workload than both Jellyfish and a Lanternfish network where every switch is connected to the same 5 rings. The result is a dynamic network capable of providing connectivity to match a wide range of workload demands, using a limited number of components.

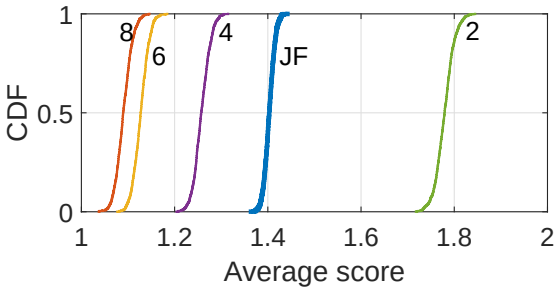


**Figure 13: 8 rings total, 2 to 8 rings connected at each switch (labelled).**

### 5.3.2 Network Locality

Our analysis so far has used uniform random workloads. We next consider a more realistic scenario, where the demand of the network exhibits locality due to the intelligent placement of applications within a datacenter. As we show here, when applications are deployed with locality along the underlying ring, dynamic Lanternfish is even better at matching the topology to the workload.

To evaluate dynamic Lanternfish for workloads with locality, we generate workload demands using a small world random distribution. In particular, we use the Watts Strogatz form of construction [24], where each node is connected to its 10 closest neighbours (5 in each direction) and with probability $p$, a connection is replaced with a globally random connection.

Figure 14 shows the effect of increasing levels of locality (smaller values of $p$) for both Jellyfish and Lanternfish, where the Lanternfish networks use 8 rings, and each switch is connected to 4 rings.

While Jellyfish remains unaffected by changes to the workload, Lanternfish's connectivity score decreases as the degree of locality increases. The improvements in connectivity is due to the decrease in connections that must travel over a large portion of the optical ring. By having more short distance connections, there is less contention for wavelengths, and makes it easier for Lanternfish to provide the required connectivity.

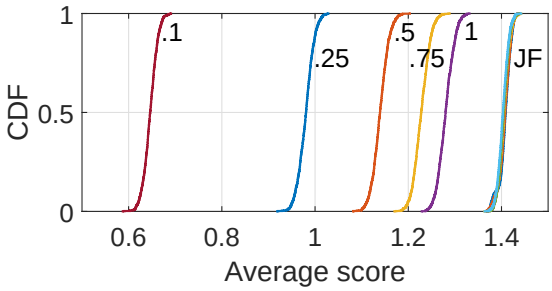


**Figure 14: Small-world random (p labelled): 8 rings total, 4 rings per switch**

### 5.3.3 *Scalability*

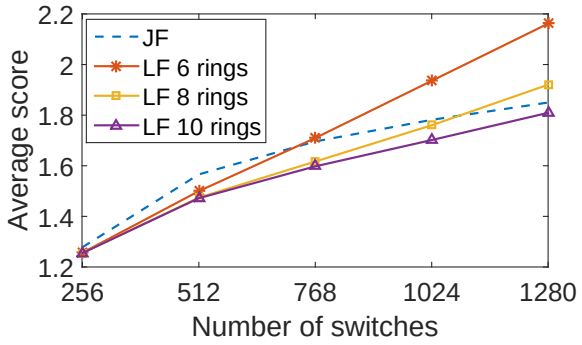


**Figure 15: Increasing network size**

A deployment of Lanternfish can vary in the number of rings that it uses. To minimize the cost of deployment, the amount of rings should depend on the size of the network, which can be described by the number of switches it contains and by the number of connections at every switch.

Figure 15 shows how well a Lanternfish network, deployed with 6, 8, and 10 rings, is able to match the workload demand across various network sizes. We vary the number of switches in the network, keep the number of connections at every switch at a constant value of 8, and use workloads with no locality. Our results are measured using our connectivity score, with the results for a Jellyfish network with twice as many transceiver shown for comparison. As the network grows, the number of rings required maintain a low score increases slowly. This type of analysis can be done by a network architect to determine the number of rings to use in order to provide the level of connectivity required for a network of a particular size.

## 6. COST ANALYSIS

To assess the affordability of deploying a Lanternfish network, we compare the cost of similarily configured Lanternfish and Jellyfish networks based on the current listed price for network components [19]. In this analysis, we do not include the cost of fiber cables because the cost of cables is negligible compared to the cost of the switching and optical components, and the required amount of cabling is dependent on the layout of the datacenter and is therefore specific to the deployment.

Figure 16 illustrates the relative cost of a static Lanternfish network compared to Jellyfish with different numbers of switch-to-switch connections. The number of ports per switch serves as a proxy for the network size, because larger random networks require more inter-switch connections in order to maintain a small network diameter. For small networks, a Lanternfish network with two rings costs approximately 3× that of a Jellyfish network. The cost increases if Lanternfish is configured with more rings; however, additional rings are only necessary for larger networks. Lanternfish becomes much more cost-competitive with large networks. It is less than 1.5× the cost of Jellyfish with only two rings, and less than 3× the cost with 6 rings. The reason that Lanternfish's relative cost decreases with larger networks is because the cost of transceivers outweighs the cost of the optical multiplexers.

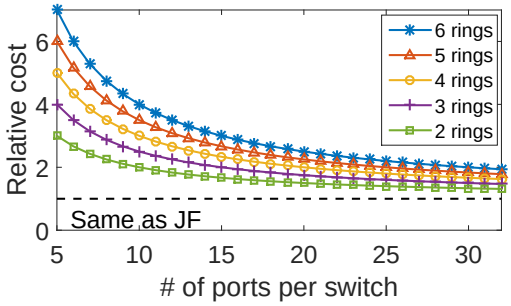


**Figure 16: Costs for static Lanternfish, relative to a Jellyfish deployment.**

Our evaluation results from Section 5 show that, for workloads that benefit from having a low hop-count between a specific set of hosts, dynamic Lanternfish can perform as well as Jellyfish while having half as many connections between switches. Therefore, for our dynamic Lanternfish cost analysis, we compare a dynamic Lanternfish network with a Jellyfish network that has twice as many interswitch connections. Figure 17 shows that, for a small network with only five ports per switch used to connect to other switches, the cost of dynamic Lanternfish is approximately 2–4× the cost of Jellyfish depending on the number of rings that are used. Surprisingly, for larger networks ($\geq$ 15 ports per switch), dynamic Lanternfish can cost less than Jellyfish using current prices. It is important to note that the cost of a dynamic Lanternfish deployment may drop significantly in the near future, assuming WSSes, which are relatively expensive, follow a similar pricing trend as optical multiplexers due to the increasing rollout of fiber-

to-the-home [25].

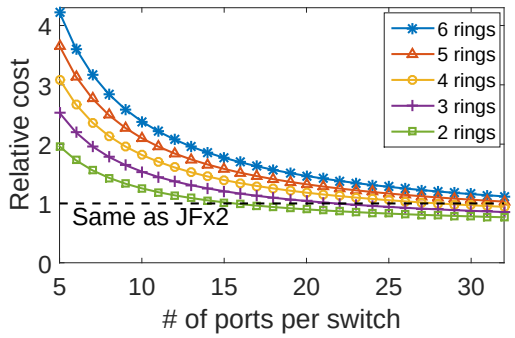


**Figure 17: Costs for dynamic Lanternfish, relative to a Jellyfish deployment with twice as many links.**

# 7. DISCUSSION

In this section, we consider additional operational challenges to deploying a random network. We also examine the feasibility of using other commodity photonic devices in constructing a datacenter network.

## 7.1 Incremental Expandability

Adding new racks to an existing Jellyfish network can be a difficult and time-consuming process. Unlike expanding a tree-network that only requires localized network changes, the new ToR switches in a Jellyfish network must connect to random endpoints dispersed across the datacenter in order to preserve uniform random connectivity. This requires undoing any previous attempts at cable management in order to remove existing connections and accommodate new connections. In contrast, expanding a Lanternfish network only requires randomly selecting wavelengths for the new ToR switches and connecting each switch to the optical rings. This is simpler and less error-prone than expanding an equivalent Jellyfish network. However, network expansions that more than double the size of the network may require additional wavelengths and rings to maintain the same network performance. This can often be avoided if future expansion is expected by initially deploying the network with more wavelengths and rings than necessary. Even when it is unavoidable, we believe that it is no more difficult or time-consuming than expanding a Jellyfish network by the same expansion factor. For example, doubling the size of an existing Jellyfish network will likely require rewiring one or more connections for every switch.

A potentially more serious problem when expanding a Lanternfish network is that it requires disconnecting one of the optical rings before reconnecting the ring to the new switch. This can lead to brief network disruptions during the expansion. Similar network disruptions occur in other random networks, although to a lesser degree, since expansion creates network-wide topology changes that will affect existing network flows. Expansion-related network disruptions for Lanternfish and other random networks can be avoided if the network supports software-defined networking (SDN) by rerouting flows before and after expansion.

## 7.2 Fault Identification

The lack of a simple deterministic structure in random networks can make identifying the source of faults more difficult than tree-based networks. A link or switch failure in a random network will likely affect connections with endpoints throughout the entire datacenter rather than in a single physical region. Pinpointing the source of faults may be simpler in Lanternfish because it should be easier to trace physical ring connections than following links (possibly traversing the entire datacenter) in a random network like Jellyfish. In either case, a more comprehensive solution is to develop tools to determine the common links and switches that would account for all of the affected endpoints.

## 7.3 Other Commodity Photonic Devices

Although wavelength selective switching components enables Lanternfish to dynamically select the mapping between transceivers and rings, it is still less flexible than topologies, such as OSA [19], that use a large optical switch to allow arbitrary connection between racks. However, by replacing standard single wavelength transceivers with transceivers having software controlled tunable lasers [26], Lanternfish can provide the same degree of flexibility as OSA. With a tunable transceiver, Lanternfish can create a connection between any arbitrary pair of switches by assigning their transceivers a wavelength that is unused by the intermediate switches between them in the ring. Although tunable transceivers are mass produced and available from multiple manufacturers, they still command a price premium over single wavelength transceivers and should only be used if the extra flexibility is necessary.

An alternative to tunable transceivers is wavelength converters, which can be built into the optical multiplexers to allow signals to switch wavelengths at each hop to avoid wavelength conflicts. An optical path between two switches is no longer restricted to a single wavelength. This can reduce the number of required wavelengths to implement a random network. However, managing wavelength conversions can be complicated and wavelength converters are not yet cost effective for use in a datacenter.

# 8. CONCLUSIONS

In this paper, we introduce Lanternfish, a new optical approach to building random datacenter networks. A

static deployment of Lanternfish greatly reduces wiring and planning complexity by implementing a random network using one or more optical rings and a random assignment of wavelengths to switches. We demonstrate that such a network provides throughput and latencies on par with Jellyfish. By adding reconfigurable optical components, we show that Lanternfish can provide better connectivity than over-provisioned static random networks, while remaining cost competitive.

## 9. REFERENCES


[1] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *NSDI*, (San Jose, CA), 2012.

[2] J.-Y. Shin, B. Wong, and E. G. Sirer, "Small-world datacenters," in *SoCC*, (Cascais, Portugal), 2011.

[3] Y. J. Liu, P. X. Gao, B. Wong, and S. Keshav, "Quartz: A new design element for low-latency DCNs," in *SIGCOMM*, (Chicago, IL), 2014.

[4] "Optical impairment-aware WSON control plane for Cisco ONS 15454 MSTP data sheet," July 2014. `http://bit.ly/RC8Ljo`.

[5] S. Miller, *Optical Fiber Telecommunications*. Elsevier, 1979.

[6] K. Grobe and M. Eiselt, *Wavelength Division Multiplexing: A Practical Engineering Guide*. John Wiley & Sons, 2013.

[7] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: Measurements & analysis," in *IMC*, (Chicago, IL), 2009.

[8] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *CoNEXT*, (Nice, France), 2012.

[9] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," in *SIGCOMM*, (Barcelona, Spain), Aug 2009.

[10] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: a scalable and fault-tolerant network structure for data centers," in *SIGCOMM*, (Seattle, Washington), Aug 2008.

[11] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient supercomputing," *IEEE Transactions on Computers*, vol. 34, no. 10, pp. 892–901, 1985.

[12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM*, (Seattle, Washington), Aug 2008.

[13] A. G. J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. M. A., P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *SIGCOMM*, (Barcelona, Spain), 2009.

[14] A. Singla, P. B. Godfrey, and A. Kolla, "High throughput data center topology design," in *NSDI*, (Seattle, WA), 2014.

[15] N. Farrington, A. Forencich, G. Porter, P.-C. Sun, J. Ford, Y. Fainman, G. Papen, and A. Vahdat, "A multiport microsecond optical circuit switch for data center networking," *Photonics Technology Letters*, August 2013.

[16] B. Lynn, P.-A. Blanche, A. Miles, J. Wissinger, D. Carothers, and L. LaComb Jr., "Design and preliminary implementation of an N x N diffractive all-optical fiber optic switch," *Journal of lightwave Technology*, vol. 31, December 2013.

[17] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *SIGCOMM*, (New Delhi, India), 2010.

[18] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in *SIGCOMM*, (New Delhi, India), 2010.

[19] K. Chen, A. Singlay, A. Singhz, K. Ramachandranz, L. Xuz, Y. Zhangz, X. Wen, and Y. Chen, "OSA: An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility," in *NSDI*, (San Jose, CA), 2012.

[20] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," in *SIGCOMM*, (Hong Kong, China), 2013.

[21] "10Gbps 100GHz DWDM SFP+ Single-Mode 40km Optical Transceiver," July 2014. `http://bit.ly/1hAwCXP`.

[22] "1RU Rack Mount, Duplex, CWDM Mux & Demux," July 2014. `http://bit.ly/1n4tOUn`.

[23] S. A. Jyothi, A. Singla, P. B. Godfrey, and A. Kolla, "Measuring and understanding throughput of network topologies," in *SIGMETRICS*, (Austin, TX), 2014.

[24] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 409–10, 1998.

[25] J. E. Berthold, "Optical networking for data center interconnects across wide area networks," in *Proceedings of the Symposium of High-performance Interconnects*, (New York, New York), August 2009.

[26] "EMCORE 10 Gbps full band tunable XFP

transceiver," July 2014. http://bit.ly/1rO4gT8.