Threshold-based Routing-Topology Co-design for Optical Data Center

Peirui Cao, Shizhen Zhao*, Dai Zhang, Zhuotao Liu, Mingwei Xu, Min Yee Teh, Yunzhuo Liu, Xinbing Wang, Chenghu Zhou

Abstract—Despite the bandwidth scaling limit of electrical switching and the high cost of building Clos data center networks (DCNs), the adoption of optical DCNs is still limited. There are two reasons. First, existing optical DCN designs usually face high deployment complexity. Second, these designs are not full-optical and the performance benefit over the non-blocking Clos DCN is not clear. After exploring the design tradeoffs of the existing optical DCN designs, we propose TROD (**T**hreshold **R**outing based **O**ptical **D**atacenter), a low-complexity optical DCN with superior performance than other optical DCNs. There are two novel designs in TROD that contribute to its success. First, TROD performs robust topology optimization based on the recurring traffic patterns and thus does not need to react to every traffic change, which lowers deployment and management complexity. Second, TROD introduces tVLB (threshold-based Valiant Load Balance), which can avoid network congestion as much as possible even under unexpected traffic bursts. We conduct simulation based on both Facebook's real DCN traces and our synthesized highly bursty DCN traces. TROD reduces flow completion time (FCT) by about 1.15-2.16× compared to Google's Jupiter DCN, at least 2× compared to other optical DCN designs, and about 2.4-3.2× compared to expander graph DCN. Compared with the non-blocking Clos, TROD reduces the hop count of the majority packets by one, and could even outperform the non-blocking Clos with proper bandwidth over-provision at the optical layer. Note that TROD can be built with commercially available hardware and does not require host modifications.

Index Terms—Optical Data Center, Reconfigurable Data Center, Routing, Topology.

1 Introduction

Traditional DCNs powered by electronic switches are facing growing bandwidth and resource demands. To cope with the demands, the data rate has increased from 10 Gbps to 40/100/200/400 Gbps in the past decade, and is expected to go even higher in the foreseeable future [2], [3], [4], [5]. However, electrical switching is becoming cost-and-energy prohibitive to keep up with the bandwidth scaling [6]. This trend has driven the development of Optical Circuit Switches (OCS) to build high-speed data centers.

However, evolving from electrically-switched DCNs to optical DCNs faces tremendous technical challenges. The de facto standard of the electrically-switched DCNs is Clos [7], [8], [9]. Due to the non-blocking structure of Clos, Clos DCNs have demonstrated superior performance. In order for optical data centers to get comparable performance, early research efforts [10], [11], [12], [13], [14], [15], [16] have proposed to reconfigure OCSs based on the time-varying traffic patterns. Nevertheless, since DCN traffic is highly bursty, even the immediate future traffic is difficult to predict. With inaccurate traffic information, the performance of the optical DCNs becomes strictly sub-optimal. Further, calculating OCS configurations is time consuming, making this design hard to react to traffic changes in real time.

To circumvent the above challenges, traffic-agnostic optical DCN design, i.e., Rotornet [17], Opera [18] or Sirius [6], was proposed. These proposals create a uniform mesh topology among ToR switches in the time-average sense by rotating through a number of pre-determined topology patterns (we thus refer to this approach as the **Rotation-based approach**), and then use valiant load balancing [19] (VLB) to handle traffic changes. These Rotation-based approaches demonstrate performance improvements over cost-comparable 3:1 oversubscribed Clos. However, in order for this approach to beat the 1:1 Clos, a completely-new co-design of switching hardware/software, host protocol stack and synchronization technology is required [6], dramatically increasing the barrier to entry. In fact, if we just apply the rotation+VLB idea on top of the existing congestion control protocol, there is still a clear performance gap from the 1:1 Clos (see Fig. 6(e) & 7(e) in §5).

Motivated by the trace studies [20], [21] that inter-point-of-delivery (PoD) traffic has certain recurring patterns, COUDER [22] and Google's Jupiter [23], [24] identify a third opportunity: low-frequency traffic-semi-agnostic PoD-level optical DCN. Instead of reconfiguring OCSs as soon as traffic pattern changes, this traffic-semi-agnostic approach optimizes DCN using multiple long-term recurring patterns. As long as a new traffic pattern is bounded by the convex hull formed by these recurring patterns, topology reconfiguration is not needed. However, although most future traffic patterns can be captured by historical traces, we may still encounter unpredictable traffic bursts. Such traffic bursts could cause severe network congestion if not properly handled. COUDER proposed a desensitization technique and Jupiter adopts a hedging approach to handle traffic bursts. However, both approaches incur bandwidth tax [18], i.e., many packets are forced to take longer paths, increasing the overall network load as a result.

To reduce bandwidth tax, while retaining the traffic-semi-agnostic design, we propose TROD. Compared with Clos, TROD's physical structure replaces all core-layer elec-

* *Corresponding author*
An earlier version of this paper appeared in ICNP 2021 [1].

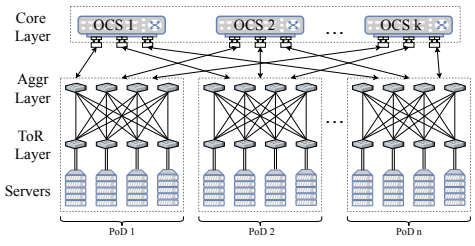


Fig. 1. Datacenter structure of TROD.

trical packet switches (EPS) by OCSs, as shown in Fig. 1. Motivated by the traffic-agnostic Valiant Load Balance (VLB) routing, TROD adopts a new threshold-based VLB (tVLB) routing, which could properly handle the unexpected traffic bursts without increasing much bandwidth tax. The basic idea is 1) when the traffic demand is below a certain threshold, traffic is routed via direct-hop/shortest paths; 2) when the traffic demand exceeds the pre-determined threshold, i.e., burst happens, traffic is load balanced to all the indirect paths, which have far more bandwidth than the direct-hop paths. By properly choosing the right threshold values based on historical traces, TROD attains efficiency by routing most traffic via direct-hop paths, while being robust to unexpected traffic bursts. When we combine TROD's OCS reconfiguration strategy with tVLB, TROD starts demonstrating superior performance.

We evaluate TROD against Clos and other DCNs using Facebook's public traces and our synthesized highly bursty traffic patterns. As expected, the 1:1 Clos offers a performance upper bound due to its rearrangeably non-blocking property. TROD performs the second best. TROD achieves 2.4-3.2$\times$ lower flow completion time (FCT) than an expander graph DCN. TROD reduces FCT by about 1.15-2.16$\times$ compared to Google's Jupiter [23], and at least 2$\times$ compared to other optical DCN designs. Since replacing a layer of electrical switches reduces cost, we also evaluate if it is possible for optical DCNs to attain better performance than the 1:1 Clos by capacity over-provisioning at the OCS layer. Our simulation results show that TROD starts outperforming the 1:1 Clos when the capacity over-provision ratio α reaches 1.2. In contrast, other optical DCN proposals either cannot beat the 1:1 Clos regardless of the over-provision ratio α, or requires a much larger value of α.

2 Background and Motivation

Full-optical DCNs, if realizable, could offer unprecedentedly higher network bandwidth than the existing electrical switching DCNs. Although there have been a number of optical DCN proposals, network vendors are still reluctant to migrate from electrical DCNs to optical DCNs. We believe that there are two main reasons:

Complexity: The adding of an OCS layer to DCN introduces a new capability of topology reconfiguration. This new capability may require new congestion control, load balancing, failure handling mechanisms, especially for frequent topology reconfiguration. Network vendors may be scared of the potential deployment and management complexities of optical DCNs, because this translates to labor and engineering costs.

Performance: None of the existing optical DCN proposals are full optical, and thus still suffer from the scaling limit of electrical switches. Since the 1:1 Clos already offers full bisection bandwidth, network vendors are unclear about the performance benefits of the existing optical DCN architectures. The only optical DCN design that claims comparable performance to the 1:1 Clos is Sirius [6]. However, Sirius requires completely new designs of optical & electrical hardware and congestion control & synchronization protocols, which dramatically increases the deployment and management complexity.

The objective of this paper is to propose a low-complexity optical DCN design with good performance. Before proposing our design, we first need to understand the following design tradeoffs of the existing optical DCNs.

2.1 Hybrid Core vs. Full Optical Core

Since DCN traffic is highly bursty and the commercially available OCSs [25], [26], [27] have a large reconfiguration delay around 30ms, the initial attempts [10], [11] used a hybrid design that routes latency-tolerant flows to the optical core and routes latency-sensitive flows to the electrical core. However, this hybrid design faces two critical problems:

1) How to accurately infer the latency requirement of different flows. Although mice flows tend to have higher latency requirement than elephant flows, this may not always be true. For example, live streaming flows are large, but are also latency sensitive.
2) How to determine the fraction between the optical core and the electrical core. Note that this number must be determined beforehand and cannot be easily changed on the fly. However, the fractions of latency-tolerant and latency-sensitive flows may change over time.

Takeaway: Determining the fraction of optical core in the hybrid architecture is difficult due to the hardness of flow classification. On the other hand, if an optical core can handle latency-sensitive traffic well, then having an electrical core may no longer be necessary.

2.2 Traffic Aware vs. Agnostic Designs

Consider optical DCNs with only optical cores. In order to handle latency sensitive traffic, the conventional wisdom [12], [13], [14], [15] is to 1) design OCSs with much lower reconfiguration latency (microsecond level); 2) and reconfigure OCSs as soon as traffic pattern changes. However, this approach encounters the following issues:

1) For OCS design, it is hard to achieve good scalability and low reconfiguration latency at the same time [16]. Although ProjectTor [15] overcame this challenge using free-space optics, the proposed optical switching technology is highly sensitive to environmental changes, and thus hard to deploy.
2) Even if OCSs can be reconfigured at very low latency, the coordination among hosts/switches/OCSs takes time, especially when the network size is large. As a result, the real time traffic pattern might have already changed

upon the completion of OCS reconfiguration. The mismatch between the OCS configurations and the current traffic pattern may deteriorate network performance (see Fig. 6(d) & Fig. 7(d)).

Due to the difficulty of handling fine grained traffic changes, [6], [17], [18] proposed a Rotation-based architecture using either rotor switch [17] or AWGR [6]. This Rotation-based architecture only requires its OCSs to be able to switch among a fixed set of configurations, and thus the optical switches used therein could achieve larger scalability without scarifying much on the reconfiguration latency. Further, a virtual uniform mesh can be formed among ToR switches in the time-average sense, and then the Rotation-based architecture can use VLB to route traffic. This design completely eliminates the necessity of traffic prediction, but introduces either deployment complexity or some performance penalty:

1) A system-wide co-design is required for the Rotation-based approach to get comparable performance to the 1:1 Clos [6], which spans switch hardware, congestion control protocol, customized synchronization protocol with an accuracy of less than 100 picoseconds[1], etc.
2) Working with the current protocol stack, the rotation + VLB idea cannot outperform the 1:1 Clos, even if we over-provision the OCS layer bandwidth (see Fig. 6(e) & Fig. 7(e)).

Another design is traffic-aware topology + traffic-agnostic routing. TAGO [29] reconfigures PoD-level topology based on traffic patterns, and performs ECMP for intra-PoD routing and topology-aware VLB for inter-block routing, i.e., for any inter-PoD packet, the probability that it chooses an intermediate PoD is proportional to the number of links between its source PoD and the intermediate PoD. However, as we will show in Fig. 6(c) and Fig. 7(c), VLB routing is far from optimal due to increased bandwidth tax.
Takeaway: Both traffic-ware and traffic-agnostic designs face many deployment complexities. The traffic-aware approaches also suffer from performance penalty due to traffic mismatch. The traffic-agnostic design might be feasible, but has a high barrier to entry.

2.3 Optical Switching over ToRs vs. PoDs

A PoD (point of delivery), with tens to hundreds of ToR switches interconnected by a number of Aggregation switches, is a basic unit for deployment [30] and incremental expansion [31] in the current commercial data centers. PoD-level optical switching has a number of advantages that might be appealing to network vendors:

1) PoD-level design agrees with the current practice that uses PoD as data center deployment unit.
2) Since a PoD is large, building a large-scale data center with over 100k servers only requires tens of PoDs. Hence, scalability is no longer an issue.
3) Due to the aggregation effect of traffic, PoD-level traffic exhibits some spatial patterns [20], [21]. This observation motivates us to design a robust OCS configuration that can handle multiple patterns.
4) It is easy to maintain connectivity between host pairs during PoD-level reconfiguration. With properly designed OCS reconfiguration steps (see §4.3), the electrical switches, the host protocol stack and the applications do not require any modification.

In contrast, researchers have generally believed that ToR-level design could deliver higher cost saving for DCN [6]. Indeed, except for a few works [10], [22], [23] that adopt a PoD-level design, more works [6], [11], [12], [13], [14], [15], [16], [17], [18] perform OCS reconfiguration over ToR switches. However, reconfiguring OCSs at the ToR layer is definitely more challenging:

1) ToR switches have small link count. Thus, to support a large scale data center with over 100k servers, thousands of ToRs would be needed. Supporting fast OCS reconfiguration over thousands of ToRs poses a scalability challenge.
2) ToR-level traffic patterns are more non-predictable [32]. As a result, the OCS controller must either 1) reconfigure OCSs as soon as it sees a newly arriving flow, which might not be feasible due to time constraint, or 2) adopt a traffic agnostic design like [6], [17], [18], whose barrier can be high in order to have comparable performance with the 1:1 Clos.
3) The connectivity between ToR switches can be intermittent. Thus, the host protocol stack might need to be modified so that it can pause/resume sending packets based on the connectivity status [11], [13].

Given the difficulty of ToR-level design, we focus on PoD-level design in this paper. At the current stage (100Gbps link speed), network cost only constitutes a small fraction of the total data center cost and a PoD-level design is much easier to implement. Admittedly, as the link speed increases to 400Gbps and beyond, network power cost may become dominant. Our design principle may also be useful for the ToR-level design. As readers will see, our design requires a certain form of traffic stability. We believe that there are two promising directions to improve the ToR-level traffic stability. First, co-design the job placement and OCS scheduling to obtain better traffic stability. Second, design optical DCN for an application with clear traffic patterns, e.g., AI training. We leave such ToR-level optical DCN designs as future works.
Takeaway: Although a PoD-level optical DCN has an additional aggregation layer as compared to a ToR-level optical DCN, it is much easier to implement, which may save significant labor and engineering cost. Further, PoD-level traffic exhibits some recurring spatial patterns, which may offer a new opportunity for better optical DCN design.

2.4 High-frequency vs. Low-frequency Reconfiguration

Most optical DCNs designs require high-frequency reconfiguration to adapt to traffic variations. They need to solve two sub-problems:
1) predicting the up-coming traffic patterns. RDC [33] and Hedera [33] predict traffic demands by the max-min fair

1. The most recent literature on data center scale time synchronization could only achieve an accuracy of tens of nanoseconds [28].

share rate of all flows within a network. Other works [34], [35], [36], [37], [38], [39], [40], [41] use machine learning models, Markov chain probability models, etc., to predict traffic patterns. However, none of the above methods can accurately predict traffic.

2) building a control plane that can quickly reconfigure the optical DCN to serve a given demand pattern. In terms of algorithm design, linear programming (LP) is often used to optimize topology and routing in reconfigurable DCNs [42], [43], [44], but LP's computational complexity is high; machine learning approaches [35], [36], [37] can also provide traffic-driven topology adaptation, but may not be robust in all situations and its cost-benefit trade-off may not be the best [45]. In terms of network control, even if the underlying optical switching hardware is fast, the software convergence speed can be slow. Note that before software convergence, the network switches may not be able to forward packets as expected. Further, even Google's state-of-art SDN controller can take several seconds to converge under certain network events [46].

Only until recently, researchers realized that topology reconfiguration frequency is not necessarily the higher the better. For example, Jupiter [23], [24] and COUDER [22] optimize DCN topologies using multiple historical traffic patterns. As a result, topology reconfiguration is not needed as long as a new traffic pattern is bounded by the convex hull formed by these historical traffic patterns. But on the other hand, both Jupiter and COUDER rely on WCMP (Weighted Cost Multi-Path) routing and incur bandwidth tax. In order to handle unexpected traffic bursts, they adopt a hedging or desensitization technique for WCMP routing to avoid forwarding traffic bursts to the same link, leading to increased average hop count and bandwidth tax.

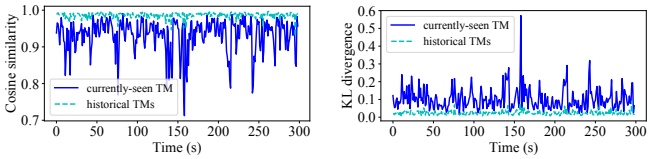
Takeaway: It is difficult to accurately predict traffic patterns and reconfiguration frequency is not necessarily the higher the better. Regarding the low-frequency reconfiguration, whether it is possible to handle traffic bursts without increasing bandwidth tax, still remains an open problem.

2.5 Understanding PoD-Level DCN Traffic

We perform a trace analysis using Facebook's public trace [21] to understand the PoD-level traffic stability. Facebook's traces were collected from three different DCN clusters (a database cluster, a web search cluster and a hadoop cluster) with a sampling rate of 1:30000. We aggregate each trace into 1-second averaged snapshots of inter-PoD traffic matrices, totaling 86400 traffic matrices in a day. Our observations are as follows.

First, PoD-level DCN traffic is not really stable. We compute the cosine similarity for every pair of adjacent TM. Fig. 2(a) plots a sequence of cosine similarity values in a 5-minute window. Clearly, PoD-level TMs can change dramatically in 1s. The cosine similarity values can be as low as 0.71, meaning that the angle between adjacent TMs can be as large as 44.8 degrees. Helios [10] proposed using the currently-seen TM to reconfigure topology. Due to the instability of DCN traffic, this approach yields poor network performance (see Fig. 6(d) & Fig. 7(d)).

Second, PoD-level DCN traffic does have a weaker form of stability, i.e., although the traffic pattern changes all the time, for any future traffic pattern, it is very likely to find a historical traffic pattern that resembles this future one. To verify this property, for every future TM, we consider all historical TMs in a 5-minute window, find the TMs that resembles this future TM most, and compute the cosine similarity between the two. We plot such similarity values in Fig. 2(a). Clearly, the similarity scores are much higher. This *weak stability* property hints that, if we could compute a DCN topology based on the set of possible historical traffic patterns, frequent reconfiguration may not be necessary.



(a) Cosine similarity (b) KL divergence.

Fig. 2. Traffic analysis using the 5-minute historical TMs vs. using the currently-seen TM.

Cosine similarity is not the only metric to analyze traffic stability. We use a different metric, Kullback-Leibler (KL) divergence, to perform the same stability analysis. KL divergence measures the distance between two TMs, and the smaller the KL divergence value, the higher the similarity between TMs. As shown in Fig. 2(b), the relative trend observed by KL divergence values is consistent with the cosine similarity values.

Takeaway: PoD-level DCN traffic varies quickly, but exhibits a weaker form of stability. Hence, we decided to perform OCS reconfiguration using a sequence of historical traffic matrices to get stronger performance guarantee (Fig. 11 & 6(a) & 7(a)). In contrast, frequent reconfiguration based on the currently-seen traffic pattern even hurts performance (Fig. 6(d) & 7(d)).

3 TROD Design

In this section, we first provide the TROD physical structure and algorithmic details of tVLB, and present how TROD utilizes tVLB to design PoD-level Topologies. Then, we prove the performance guarantee of TROD.

3.1 TROD's Physical Structure

After exploring the design tradeoffs of existing optical DCN proposals, we propose TROD, a high-performance optical DCN with low deployment complexity. The network architecture is shown in Fig. 1. The PoDs are all connected to the OCS layer. Note that an OCS is a fully optical component that sends incoming optical signals directly to a reconfigurable egress port without packet decoding. Although OCS reconfiguration takes time, upon completion of OCS reconfiguration, a new inter-PoD topology is formed and OCSs become transparent to in-flight packets. In the rest of this paper, we refer to the process of changing inter-PoD topology by OCS reconfiguration as **Topology Engineering (ToE)**. Both Helios [10] and TROD perform ToE in the PoD layer, but TROD differs from Helios in the following aspects:

1) TROD's architecture is simpler than Helios: 1) an additional electrical core is not needed; 2) mice-elephant classification methods are not needed.
2) Unlike Helios, TROD does not react to every traffic matrix (TM) change. To achieve this goal, TROD's routing (§3.2) and topology (§3.3) are both designed based on the long-term traffic characteristics extracted from the historical TMs, and are optimized to be robust against traffic uncertainty.
3) TROD has much lower deployment and management complexity. Compared to Helios, TROD's reconfiguration frequency can be much lower. Notably, our simulation in §5 suggests that daily reconfiguration is already good enough for Facebook's public DCN traces. With such a low reconfiguration frequency, the optical DCN is almost static. Thus, the existing control and management strategies for static DCNs, including congestion control, failure handling, etc., still work for TROD.

We use a sequence of TMs $D(t) = [d_{ij}(t), i, j = 1, ..., n]$, ($n$ is the number of PoDs), to compute the PoD-level topology $X = [x_{ij}]$, where x_{ij} is the number of links between PoD i and PoD j. This topology X must satisfy the following physical constraints:

$$\begin{cases} \sum_{j=1}^{n} x_{ij} \le r_i, \sum_{i=1}^{n} x_{ij} \le r_j, \forall i, j = 1, ..., n \\ x_{ij} \text{ are non-negative integers and } x_{ii} = 0, \end{cases} \quad (1)$$

where r_i is the number of bidirectional links between PoD i and the OCS layer.

TROD's objective is to design a topology solution that minimizes the worst-case link congestion for future TMs. Clearly, routing protocols also affect the topology design and the final network performance. We have tried the widely used ECMP and VLB routing protocols, but unfortunately did not obtain good performance. Finally, we propose a new routing protocol, called threshold-based VLB (tVLB).

3.2 TROD's tVLB Routing

Given an inter-PoD topology $X = [x_{ij}, i, j = 1, 2, ..., n]$, tVLB sets a data rate threshold $s_{ij} \le C_{ij}$ for every traffic component d_{ij}, where $C_{ij} = Bx_{ij}$ is the link capacity of PoD pair (i, j). Then, tVLB routes DCN traffic $D = [d_{ij}]$ as follows:

- If the traffic demand $d_{ij} \le s_{ij}$, then all the traffic from PoD i to PoD j will be routed to the direct-hop path $i \to j$.
- If the traffic demand $d_{ij} > s_{ij}$, then s_{ij} amount of traffic will still be routed to the direct-hop path $i \to j$, while the excessive traffic $d_{ij} - s_{ij}$ will be routed to all the two-hop paths $i \to k \to j, k \ne i, j$, based on the following routing weights

$$\gamma_{i(k)j} = \frac{\min\{C_{ik} - s_{ik}, C_{kj} - s_{kj}\}}{C_{ij}^{\text{2hop}}}, k \ne i, j, \quad (2)$$

where $\min\{C_{ik} - s_{ik}, C_{kj} - s_{kj}\}$ is the available two-hop capacity along the path $i \to k \to j$ (note that s_{ij} amount of capacity has been reserved for direct-hop routing), and $C_{ij}^{\text{2hop}} = \sum_{l \ne i,j} \min\{C_{il} - s_{il}, C_{lj} - s_{lj}\}$ is the total amount of two-hop capacity. Clearly, $\sum_{k \ne i,j} \gamma_{i(k)j} = 1$.

tVLB can be viewed as a "traffic aware" version of VLB. In fact, if the residual topology $[C_{ij} - s_{ij}]$ is a perfect uniform mesh, then all the $\gamma_{i(k)j}$'s would be equal, which aligns with the routing weights of VLB. If we set all the thresholds as 0, then tVLB degenerates to VLB. Note that tVLB is used to deal with inter-PoD traffic. We use ECMP to deal with intra-PoD traffic because intra-PoD is a static Clos structure.

Why choose tVLB: At the beginning, we believed that after performing ToE, all the traffic could be simply routed along the shortest paths using ECMP. However, due to the unexpected traffic bursts, ECMP may cause severe network congestion and dramatically increase the flow completion time (see Fig. 6(b) & 7(b)). In order to mitigate the impact of traffic bursts, we then tried VLB. However, VLB will route a majority of traffic through non-shortest paths. This increases the overall network load as well as the FCT (see Fig. 6(c) & 7(c)). tVLB combines their benefits. With proper thresholds, the majority of traffic can be still routed to the shortest paths, while the unexpected traffic bursts can be load balanced among non-shortest paths to avoid congestion.

3.3 Detailed Design of TROD's Topology under tVLB

General idea of ToE under tVLB routing: Clearly, we should setup more links between hot PoD pairs. We can first compute the p-th percentile value $s_{ij}(p)$ for d_{ij} based on its historical trace $d_{ij}(t), t_1 < t < t_2$, and then use $s_{ij}(p)$ as its tentative threshold for tVLB. Note that $s_{ij}(p)$ acts as the value of traffic prediction. Obviously, we need to make sure $Bx_{ij} > s_{ij}(p)$. Since tVLB uses VLB to route traffic that exceeds its corresponding threshold, we should design the residual topology $Bx_{ij} - s_{ij}(p)$ as uniform as possible. Then, the basic formulation for ToE becomes

$$\begin{aligned} \max_{X=[x_{ij}]} \quad & \Delta \\ \text{s.t.} \quad & X \text{ satisfy (1) and } Bx_{ij} - s_{ij}(p) \ge \Delta, \forall i, j. \end{aligned} \quad (3)$$

In practice, due to many physical constraints and the imbalance of DCN traffic, it may not always be possible to obtain a completely uniform residual topology. To deal with this issue, we use Progressive Filling (Alg. 1) to achieve a max-min fairness allocation for the PoD-level topology.

The input of Alg. 1 is a percentile value p and a time sequence of historical TMs $D(t) = [d_{ij}(t), i, j = 1, 2, ..., n], t_1 < t < t_2$. The output is TROD's topology and routing thresholds. Note that lines 8-18 show the pseudo code for topology calculation. The core idea is progressive filling. We first allocate $\lfloor \frac{s_{ij}(p)}{B} \rfloor$ number of links to the pod pair (i, j), and then increase the allocation uniformly until some physical constraints in (1) change from $<$ to $=$. In line 12-17, we mark the (i, j) pairs contained in equality constraints as "done". The progressive filling terminates until all (i, j) pairs are done with allocation.

The progressive filling algorithm cannot guarantee that every (i, j) pair is allocated with capacity higher than its initial threshold $s_{ij}(p)$. This is because the x_{ij}'s must be integers and some $s_{ij}(p)$'s might be too small to be allocated

Algorithm 1: Progressive Filling Algorithm

Data: A percentile value p, and a time sequence of historical TMs $D(t) = [d_{ij}(t), i, j = 1, 2, ..., n], t_1 < t < t_2$.

Result: Inter-PoD topology $X = [x_{ij}, i, j = 1, 2, ..., n]$, and routing thresholds $S = [s_{ij}, i, j = 1, 2, ..., n]$.

// Initialization

1 Define a link margin η and initialize $\eta = 0$.

2 Define a set Ω to track the (i, j) entries that are already done with allocation, and initialize $\Omega = \{(i, i), i = 1, 2, ..., n\}$.

3 For every $i \neq j$, Set $s_{ij}(p)$ as the p-th percentile value of $d_{ij}(t), t_1 < t < t_2$.

4 **if** $\sum_{k=1}^{n} s_{ik}(p) > Br_i$ *or* $\sum_{k=1}^{n} s_{ki}(p) > Br_i$ *for some* i **then**

5 Raise an alert to reduce p or upgrade PoD i.

6 exit()

7 **end**

8 Initialize $x_{ij} = \lfloor \frac{s_{ij}(p)}{B} \rfloor$ for all $i, j = 1, 2, ..., n$.

// Calculate topology

9 **while** $\Omega \neq \{(i, j), i, j = 1, 2, ..., n\}$ **do**

10 Find the smallest η such that there exists an $(i, j) \notin \Omega$ satisfying $\frac{s_{ij}(p)}{B} + \eta \geq x_{ij} + 1$, and pick one such (i, j).

11 Increase x_{ij} by 1.

12 **if** $x_{i1} + x_{i2} + \cdots + x_{in} == r_i$ **then**

13 $\Omega = \Omega \cup \{(i, 1), (i, 2), ..., (i, n)\}$

14 **end**

15 **if** $x_{1j} + x_{2j} + \cdots + x_{nj} == r_j$ **then**

16 $\Omega = \Omega \cup \{(1, j), (2, j), ..., (n, j)\}$

17 **end**

18 **end**

// Set up routing threshold

19 Let $\eta^* = \min_{i,j,Bx_{ij} > s_{ij}(p)} \{Bx_{ij} - s_{ij}(p)\}$

20 Set $s_{ij} = \begin{cases} Bx_{ij}, & \text{if } Bx_{ij} \leq s_{ij}(p) \\ Bx_{ij} - \eta^*, & \text{if } Bx_{ij} > s_{ij}(p) \end{cases}$

21 return $X = [x_{ij}]$ and $S = [s_{ij}]$;

a link. Besides, some (i, j) pairs may be allocated with capacity much higher than their corresponding $s_{ij}(p)$'s. This could happen when both PoD i and PoD j are lightly loaded. In this situation, we may increase the threshold for such (i, j)'s to allow more traffic going direct-hop paths. These two cases are accommodated in lines 19-20 of Alg. 1.

Remark on the Algorithmic Complexity: Alg. 1 use a while loop (see lines 9-18) to perform bandwidth allocation. This while loop can be executed at most $\sum_{i=1}^{n} r_i$ times, because each iteration increases the number of allocated links by one and the total number of links cannot exceed $\sum_{i=1}^{n} r_i$. Each iteration requires $\Theta(n^2)$ operations. Thus, the overall algorithmic complexity is $\Theta(n^2(\sum_{i=1}^{n} r_i))$. In contrast, the topology calculation algorithm adopted by either COUDER [22] or Google's Jupiter DCN [23] [24] requires solving three linear programming problems with $\Theta(n^3)$ variables, and thus the overall computational complexity is much higher.

Remark on the Parameter p: Alg. 1 takes p as its input. In practice, we can optimize the choice of p based on the burstiness of the DCN TMs. If the DCN traffic is highly busrty, a smaller value of p is preferred, because more non-shortest path bandwidth is reserved to combat traffic bursts; otherwise, a larger value of p is preferred, because more traffic can take the shortest paths. To achieve the above goal, one can try multiple values of p, use Alg. 1 as a subroutine to compute a topology/routing solution for each p, and then evaluate each topology/routing solution based on certain design requirements to determine the best value of p.

3.4 Performance Guarantee of TROD

To understand the performance of TROD, we characterize an inner bound of TROD's capacity region[2] as follows:

Theorem 1. Given TROD's topology solution $X = [x_{ij}]$ and routing thresholds $S = [s_{ij}]$, if a traffic matrix $D = [d_{ij}]$ satisfy the following constraints:

$$\sum_{k \neq i,j} \left(\frac{(d_{ik} - s_{ik})^+}{C_{ik}^{\text{2hop}}} + \frac{(d_{kj} - s_{kj})^+}{C_{kj}^{\text{2hop}}} \right) \leq 1, \forall i \neq j, Bx_{ij} > s_{ij}, \tag{4}$$

then D can be supported by TROD, i.e., the max link utilization (MLU) of routing D over TROD is no more than 1.

Theorem 1 offers a sufficient Condition (4) for a TM D to be supportable by TROD under tVLB routing. This condition defines a convex set for D. Clearly, given the same thresholds, larger two-hop capacity values could help enlarge the above convex set, and thus make the DCN more robust to traffic bursts. (Readers can interpret $(d_{ij} - s_{ij})^+ = \max\{d_{ij} - s_{ij}, 0\}$ as the burst component of d_{ij}.) TROD achieves as large two-hop capacity values as possible by equalizing $[Bx_{ij} - s_{ij}]$ for different (i, j) pairs based on max-min fairness.

Proof 1. Consider an arbitrary link (i, j). The traffic traversing this link can be grouped into three categories:

1) Traffic sent from PoD i to PoD j through direct hop, which equals $\min\{s_{ij}, d_{ij}\} \leq s_{ij}$;
2) Traffic sent from PoD i to PoD k through PoD j, which equals $(d_{ik} - s_{ik})^+ \gamma_{i(j)k} \leq (d_{ik} - s_{ik})^+ \frac{Bx_{ij} - s_{ij}}{C_{ik}^{\text{2hop}}}$;
3) Traffic sent from PoD k to PoD j through PoD i, which equals $(d_{kj} - s_{kj})^+ \gamma_{k(i)j} \leq (d_{kj} - s_{kj})^+ \frac{Bx_{ij} - s_{ij}}{C_{kj}^{\text{2hop}}}$.

Then, the total amount of traffic on the link (i, j) is upper bounded by

$$s_{ij} + (Bx_{ij} - s_{ij}) \sum_{k \neq i,j} \left(\frac{(d_{ik} - s_{ik})^+}{C_{ik}^{\text{2hop}}} + \frac{(d_{kj} - s_{kj})^+}{C_{kj}^{\text{2hop}}} \right). \tag{5}$$

According to line 20 of Alg. 1, $Bx_{ij} \geq s_{ij}$. If $Bx_{ij} > s_{ij}$, then (5) $\leq s_{ij} + (Bx_{ij} - s_{ij}) = Bx_{ij}$. If $Bx_{ij} = s_{ij}$, then (5) $= s_{ij} = Bx_{ij}$. In either case, the link utilization of (i, j) is no higher than 1. Q.E.D.

2. Capacity region is defined as the closure of the set of all possible traffic matrices that can be stably supported by a network.

This proof and Alg. 1 imply two reasons for TROD's good performance:

1) Traffic-similar base topology can deal with common traffic and the residual uniform topology conquers bursty traffic.
2) tVLB only allows excessive traffic to go through non-shortest paths. Thus, it greatly reduces the average hop count or bandwidth tax when compared with the routing approaches adopted by COUDER [22] and Google's Jupiter DCN [23], [24].

Equation (4) offers a sufficient condition for MLU to be less than 1. We validate this condition numerically in Fig. 3. We randomly pick a logical topology $X = [x_{ij}]$ and set the threshold value s_{ij} as $0.8Bx_{ij}$. We generate 1000 random TMs. For each TM, we calculate the Left-Hand Side (LHS) value of (4) and the max link utilization (MLU) under tVLB. The results in Fig. 3 indicate that as long as the LHS value of (4) is less than or equal to 1, the MLU must be less than 1. Even when the LHS value of (4) is greater than 1, the MLU can be still less than 1. In our 1000 experiments, the LHS value of (4) can be as large as 5. But only 16 of them have an MLU value greater than 1.

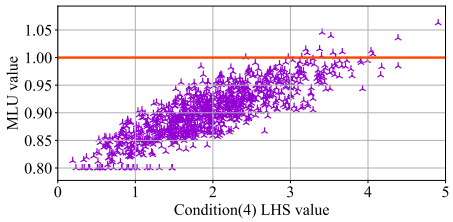


Fig. 3. Numerical analysis experiment.

4 Implementation

This section describes TROD's implementation details. TROD does not require any customized switch hardware, including electrical switches and optical circuit switches. No modifications to the host or application are required.

4.1 Realizing Inter-PoD Topology

Equation (1) defines the physical constraints between the gigantic aggregation switches and one gigantic core-layer OCS. When deployed on large-scale data centers, the physical constraints between multiple aggregation switches and multiple OCSs must be considered [47]. TROD computes an inter-PoD topology $X = [x_{ij}]$ based on historical traffic matrices. However, this topology solution does not reflect the following physical constraints:

1) There are multiple aggregation switches in a PoD.
2) There are multiple OCSs in the core layer.

Here, we describe how to map TROD's PoD-level topology $X = [x_{ij}]$ to the underlying switch hardware and discuss its implications to tVLB routing.

Similar to the design of Google's Jupiter data centers [8], [23], [24], we assume that each PoD has M identical aggregation switches. We design the physical connections between the aggregation switches and the OCSs based on the following principles:

1) There are M groups of OCSs, and the m-th aggregation switch only connects to the m-th OCS group.
2) In each OCS group, every aggregation switch spreads its uplinks evenly across all the OCSs.

Given the above physical topology, in order to realize a PoD-level topology $X = [x_{ij}]$ using OCSs, we need the following Theorem.

***Theorem* 2.** Given an $I \times J$ non-negative interger matrix $X = [x_{ij}]$, then for any integer $M \geq 1$, there exist M non-negative integer matrices $X(1), ..., X(M)$ satisfying
1. $X = X(1) + ... + X(M)$;
2. for any $i = 1, ..., I, j = 1, ..., J$ and $m = 1, 2, ..., M$, $\lfloor \frac{x_{ij}}{M} \rfloor \leq x_{ij}(m) \leq \lceil \frac{x_{ij}}{M} \rceil$;
3. for any $i = 1, ..., I, j = 1, ..., J$ and $m = 1, 2, ..., M$, $0 \leq \sum_{j=1}^{J} x_{ij}(m) \leq \lceil \frac{\sum_{j=1}^{J} x_{ij}}{M} \rceil$;
4. for any $i = 1, ..., I, j = 1, ..., J$ and $m = 1, 2, ..., M$, $0 \leq \sum_{i=1}^{I} x_{ij}(m) \leq \lceil \frac{\sum_{i=1}^{I} x_{ij}}{M} \rceil$;

Theorem 2 is a special case of the Theorem 3 in Appendix A.1 of [31]. It can be proved by transforming the matrix decomposition process into a sequence of max-flow problems. See Appendix A.2 in [31] for the detailed proof. Next, we describe how to realize $X = [x_{ij}]$ in two steps.

Step 1: Decompose X into M sub-topologies $X(m) = [x_{ij}(m)], m = 1, ..., M$, each of which corresponding to an OCS group. Recall that there are M groups of OCSs in the DCN core layer and M aggregation switches in each PoD, and the m-th aggregation switches in different PoDs only connect to the m-th OCS group. The m-th sub-topology $X(m)$ characterizes the topology among the m-th aggregation switches in all PoDs. In our multi-OCS physical structure, the total number of uplinks between each PoD and each OCS group is exactly $\frac{r_i}{M}$, because the M aggregation switches in each PoD are identical. According to Theorem 2, it is always feasible to find M sub-topologies $X(m), m = 1, ..., M$, such that

$$\begin{cases} x_{ij} = \sum_{m=1}^{M} x_{ij}(m), \forall i, j = 1, ..., n, \\ \sum_{j=1}^{n} x_{ij}(m) \leq \frac{r_i}{M}, \forall i = 1, ..., n, m = 1, 2, ..., M, \\ \sum_{i=1}^{n} x_{ij}(m) \leq \frac{r_j}{M}, \forall j = 1, ..., n, m = 1, 2, ..., M, \\ x_{ij}(m) \text{ are integers in } [\lfloor \frac{x_{ij}}{M} \rfloor, \lceil \frac{x_{ij}}{M} \rceil]. \end{cases} \tag{6}$$

Step 2: For each $m = 1, ..., M$, decompose $X(m)$ into K sub-topologies $X(m, k) = [x_{ij}(m, k)], k = 1, ..., K$, each of which corresponding to an OCS in the m-th OCS group. Here, we have assumed that each OCS group has K OCSs. Recall that in each OCS group, every aggregation switch spreads its uplinks evenly across all the OCSs. The per-OCS sub-topology $X(m, k)$ can be computed using a similar approach as (6):

$$\begin{cases} x_{ij}(m) = \sum_{k=1}^{K} x_{ij}(m, k), \forall i, j = 1, ..., n, \\ \sum_{j=1}^{n} x_{ij}(m, k) \leq \frac{r_i}{MK}, \forall i = 1, ..., n, k = 1, 2, ..., K, \\ \sum_{i=1}^{n} x_{ij}(m, k) \leq \frac{r_j}{MK}, \forall j = 1, ..., n, k = 1, 2, ..., K, \\ x_{ij}(m, k) \text{ are integers in } [\lfloor \frac{x_{ij}(m)}{K} \rfloor, \lceil \frac{x_{ij}(m)}{K} \rceil]. \end{cases} \tag{7}$$

Remark on the Decomposition Process: The above decomposition process requires solving MK max flow problems with $\Theta(n)$ nodes. A Max-flow problem can be solved using

the Goldberg-Tarjan algorithm [48] with $\Theta(n^3)$ complexity. Thus, the overall algorithmic complexity of the decomposition process is $\Theta(n^3MK)$. Thanks to this decomposition process, TROD only needs to care about designing a PoD-level topology over a single gigantic core-layer OCS. This not only reduces TROD's algorithmic complexity, but also makes it easier to optimize performance. Indeed, it was shown in [49] that with only one gigantic OCS, optimal network topologies can be computed in polynomial time.

4.1.1 Realizing tVLB with Multiple Aggregation Switches

TROD sets up a routing threshold s_{ij} for every PoD pair (i, j). Next, we present how to realize tVLB with multiple aggregation switches in a PoD.

For every packet originated from PoD i, it traverses a ToR switch and one of the M aggregation switches before leaving PoD i. We assign IP addresses to servers and switches based on different IP prefixes in different PoDs. Hence, IP prefixes can be used to identify the destination of a packet. Since M aggregation switches are disjoint, load balancing must be performed at every ToR switch. Based on TROD's design, the majority of traffic of d_{ij} will go directly from PoD i to PoD j. Hence, we set up the weighted cost multipathing (WCMP) [50] weights proportional to $x_{ij}(1) : x_{ij}(2) : \cdots : x_{ij}(M)$ for the traffic d_{ij} at each ToR.

We then set up routing thresholds for tVLB at every aggregation switch. Based on the WCMP weights at ToR switches, the threshold of the demand d_{ij} at the m-th aggregation switch should be configured as $s_{ij} \cdot \frac{x_{ij}(m)}{\sum_{m=1}^{M} x_{ij}(m)}$.

4.2 Implementing tVLB

TROD uses tVLB for routing. tVLB can be supported with OpenFlow 1.3 [51] switches or P4 [52] switches.

4.2.1 Implementing tVLB on OpenFlow 1.3 switches.

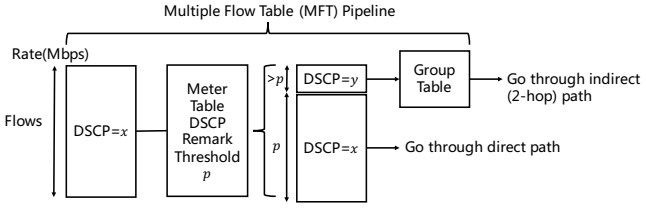


Fig. 4. Flow chart of tVLB using meter table and MFT pipeline.

We first discuss an implementation using OpenFlow 1.3 switches. Openflow 1.3 switches support two key features to realize tVLB: meter table [51] and multiple flow table (MFT) pipeline. Meter table can measure the packet rate and perform some rate-related packet operations. MFT allows packet processing in multiple stages. As shown in Fig. 4, tVLB can be implemented as follows:

Step 1: Define a meter table with band type as *dscp remark*, and use the desired threshold p as the rate limit. *dscp remark* modifies the DSCP field in the IP header of a packet, which is used to define a simple DiffServ Policy. We set the default value of DSCP as x (usually 0). The meter table measures the packet rate, and modifies the DSCP value[3] to y for all the packets that exceed the rate limit p.

Step 2: Set a flow rule in *Table 0* that matches the desired fields, e.g., the source PoD and the destination PoD's ip prefixes. All the matched packets are first directed to the above meter table, and then sent to *Table 1*.

Step 3: Set two flow rules in *Table 1* that perform different forwarding actions based on DSCP values. Packets with unmodified DSCP values are forwarded to the direct-hop path. Packets with modified DSCP values are forwarded to a *Group Table*, which hashes flows to different indirect paths. Note that packets of the same flow are hashed to the same path. This help reduce the amount of out-of-order delivery.

We have tested the above design in *ofsoftswitch13* [53], and thus any software or hardware switch that fully supports OpenFlow 1.3 [54], [55], [56], [57] should be able to implement tVLB.

4.2.2 Implementing tVLB on P4 switches

Unfortunately, most commercial switches do not support full OpenFlow 1.3 functionality [58], [59], [60]. Therefore, to make TROD more widely deployed, we further research how to realize tVLB on P4 switches, which have better prospects and support more flexible programming. We implement a tVLB prototype and make it open-source [61].

We choose the Wedge100BF-32X switch equipped with the Barefoot Tofino chipset as our P4 hardware, and use the P4 capabilities including reconfigurable Match-Action Table (MAT) pipeline and meters to implement tVLB.

Our P4 switch supports hardware-implemented Two Rate Three Color Marker (trTCM) meters which can mark packets either one of 3 values (green, yellow, or red) given two thresholds, i.e. Peak Information Rate (PIR) and Committed Information Rate (CIR), as the meter's configuration. A packet is marked red if it exceeds the PIR, otherwise it is marked either yellow or green depending on whether it exceeds the CIR. Here we only need one threshold to tell the traffic apart so we use the PIR as the rate limit. The implementation of tVLB in P4 is described as follows:

Step 1: Create a meter rule and a forwarding rule, and select the *ipv4.diffserv* mode for both rules. The packets belonging to the same source and destination PoD's IP prefixes are measured by the same meter rule.

Step 2: Modify the DSCP value of a packet to y if this packet is marked red.

Step 3: The forwarding table checks the DSCP value of all the packets and forwards the packets to different paths accordingly. Packets with unmodified DSCP values are forwarded to the direct-

3. According to RFC 4594, CS7 (DSCP value = 56) is currently unused. We can use CS7 to impment tVLB.

hop path. Packets with modified DSCP values are forwarded to the indirect-hop paths.

We have verified that our implementation of tVLB on P4 switches is capable of splitting the traffic according to the threshold value, and we have benchmarked the accuracy of rate control varying the rate limit configurations as well as ingress traffic volumes. In addition, implementing tVLB in data plane only consumes a small amount of hardware resources. The detailed results are shown in §5.7.

Remark on out-of-order delivery: tVLB routing may cause out-of-order delivery of packets, as a flow may switch between a direct-hop path and an indirect-hop path occasionally. Fortunately, we can migrate the problem with existing protocols. More details are available in §5.4.

Another approach to implement tVLB: We can also use P4 Metadata to implement tVLB. Specifically, we can mark a flag in the packet's Metadata based on the traffic rate and then split the traffic according to this flag. Compared to the DSCP based approach, using Metadata does not conflict with the normal usage of DSCP, but on the other hand, metadata may not be supported in other types of switches.

4.3 Reconfiguring OCSs and Switches

TROD's routing and topology solutions are designed using a sequence of historical traffic and are optimized against traffic bursts. As a result, TROD does not have to perform frequent reconfiguration to react to demand variations. Then, TROD does not need to rush for reconfiguration, and can put safety as its primary goal during reconfiguration.

To avoid routing packets to black holes, TROD uses logical ports to set up flow rules in the aggregation switches. A logical port can be either a trunk port or a link aggregation group, which can be configured to contain an arbitrary set of physical ports. While physical connections between two aggregation switches could change upon reconfiguration, the logical port id for every switch pair remains unchanged. Hence, during reconfiguration, as long as every active logical port contains at least one physical port, blackholing can be avoided.

To avoid losing capacity during reconfigurations, especially when the network load is high, TROD performs reconfigurations in multiple steps. To reduce the number of reconfiguration steps, TROD adopts minimal rewiring [31] to reduce the total number of links to be reconfigured.

After OCS reconfiguration, TROD can then update every flow/meter rule with new threshold values and new routing weights. Note that, we do not need to modify applications or host protocol stacks for TROD's reconfiguration process.

5 Performance Evaluation

We evaluate TROD against different DCNs. The baseline is Clos. Depending on the volume of network traffic, network vendors may deploy either oversubscribed Clos or non-oversubscribed Clos. Hence, we will evaluate both 1:1 (non-oversubscribed) Clos and 2:1 (oversubscribed) Clos.

In our simulation setup, the topology of TROD is shown in Fig. 1, which replaces all core-layer electric packet switches (EPS) with optical circuit switches (OCS). Each ToR EPS has 64 ports with 10 Gbit/s port rates. The number of aggregation-layer EPS ports depends on the over-provision ratio α. The link latency between different PoD pairs is set as 600ns, and the link latency between the ToR switch and the aggregation switch inside a PoD is set as 20ns. For congestion control, we use DCTCP [62].

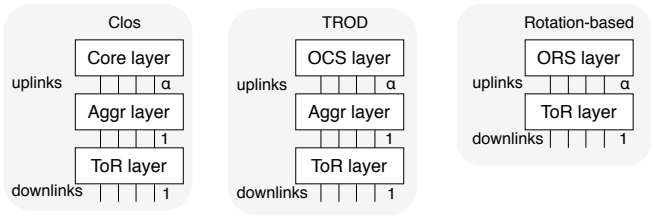


Fig. 5. Illustrating over-provision ratio α over different DCN architectures.

Metric: The primary performance metric is **Flow Completion Time (FCT)**. FCT is closely related to user experience, and thus is probably the most important performance metric for users [63]. Since FCT is hard to compute mathematically, we use our packet-level simulator [61], which is extended from an open source network simulator NetBench [64]. Note that Facebook's traces contain flows of different sizes. To allow better comparison across flows and DCNs, we use ***FCT slowdown*** [30], which is a flow's actual FCT normalized by its ideal FCT when the network only has this flow.

Another performance metric is **Max Link Utilization (MLU)**. MLU measures the worst congestion level across all the links in the DCN, and is widely used by network operators to monitor their DCN fabrics.

Over-Provision Ratio: User experience is the key to success for cloud providers. If migrating from Clos to optical DCN hurts network performance, network vendors may not be willing to give a try. The 1:1 Clos is rearrangably non-blocking, offering excellent network performance. Then, a natural question arises: is it possible for optical DCN to get comparable or even better performance than Clos?

Note that the number of hops of the shortest paths in optical DCNs is fewer than that of the Clos DCN, and that the unit price of an OCS port is typically cheaper than that of an electrical switch port. If we over-provision the OCS layer capacity, the optical DCN may achieve better performance than the 1:1 Clos. For ease of evaluation, we introduce *Over-Provision Ratio*, denoted by α, which is equal to the total core-layer uplink capacity divided by the total ToR-layer downlink capacity. This concept is also illustrated in Fig. 5 for different DCN architectures.

5.1 TROD vs. Traffic-aware and Traffic-agnostic DCNs

We evaluate FCT slowdown using Facebook's production traces [21] for the following DCN architectures:

Traffic-semi-aware optical DCN (TROD): TROD uses tVLB routing by default. We also evaluate ECMP and VLB for TROD. For any of the three routing options, four over-provision ratios, $1, 1.2, 1.4$ and 2 are evaluated. If not stated otherwise, daily reconfiguration is used.

Traffic-aware optical DCN [10]: Use the currently-seen traffic matrix to perform Pod-level reconfiguration, and then use ECMP for routing. Four over-provision ratios, $1, 1.2, 1.4$ and 2 are evaluated.

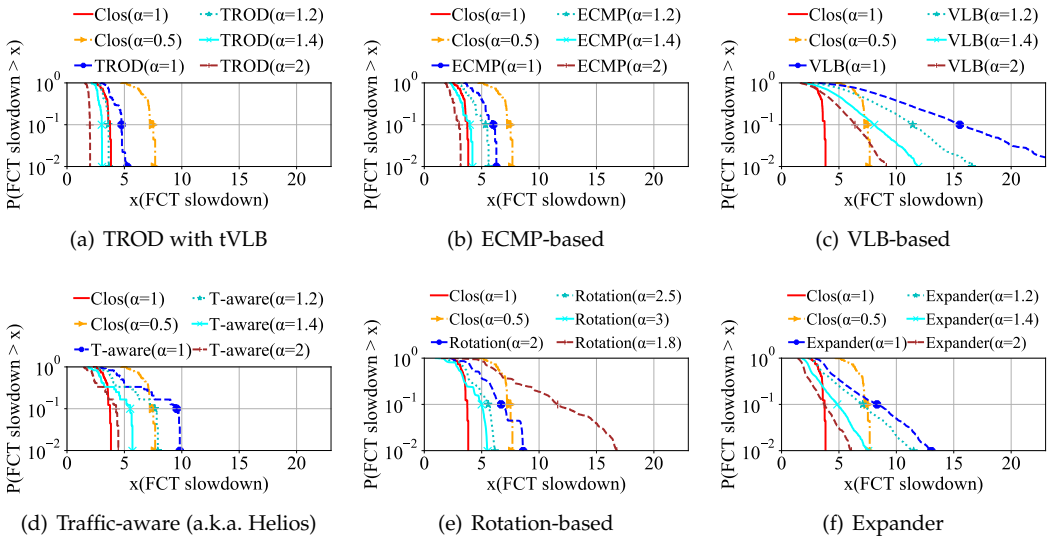


Fig. 6. Performance under common traffic patterns. P(FCT slowdown>x) is the probability that FCT slowdown exceeds x. Clos($\alpha = 1$) is non-oversubscribed. Clos($\alpha = 0.5$) is oversubscribed. Traffic-aware approach reconfigures topology every second based on the currently-seen traffic.

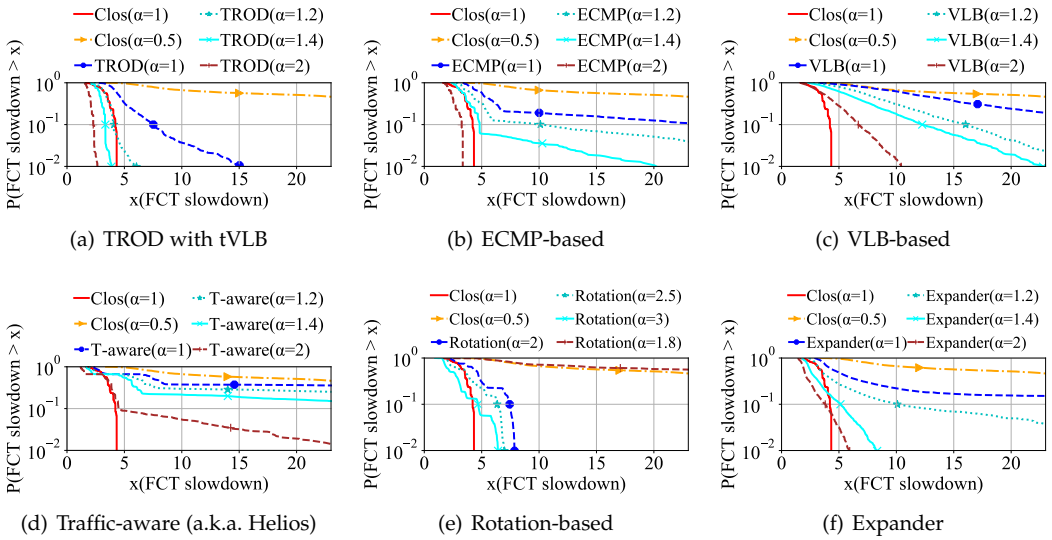


Fig. 7. Performance under synthetic bursty traffic patterns.

Traffic-agnostic Rotation-based optical DCN [6], [17], [18]: Rotate OCS configurations every 100ns, and then use VLB for routing. [6] has shown that this approach can achieve comparable performance with the 1:1 Clos with customized switches, hosts, and congestion control protocols. Here, we are curious about its performance without any customization. Four over-provision ratios, $1.8, 2, 2.5$ and 3 are evaluated. (This approach does not have an aggregation layer, and thus larger α can be used without incurring higher cost.)

Expander graph DCN [65], [66]: We use a static uniform mesh topology to simulate a performance upper bound for the expander graph. A uniform mesh topology is an expander with the optimal edge expansion. K-shortest path routing is used for this expander graph. Four over-provision ratios, $1, 1.2, 1.4$ and 2 are evaluated.

For the traffic-aware approach and the Rotation-based approach, we set the OCS reconfiguration latency as 0 in our simulation. Thus, all the results we obtain for the two approaches are actually performance upper bounds.

5.1.1 *Evaluating Common Traffic Patterns*

To capture the diurnal patterns of Facebook's trace, we pick different trace segments from different times of a day, and simulate these trace segments one by one. We collect FCTs for all the finished flows, compute the FCT slowdown values, and plot the results in Fig. 6.

Clearly, TROD with tVLB routing performs the best. It starts outperforming the 1:1 Clos from $\alpha = 1.2$. When $\alpha = 1.4$, TROD is strictly better than the 1:1 Clos, reducing FCT by about $1.3\times$. The key to TROD's success is that, TROD can route the majority of packets through direct-hop paths, while ensuring the direct-hop paths are not congested. Note that packets have to traverse one more hop in Clos.

The second-best option is TROD with ECMP routing. In this case, even if all the packets take direct-hop paths, due to link congestion, the resulting FCT slowdown turns out to be 1.2-1.6× larger when compared to the default TROD. Further, TROD with ECMP also requires a larger over-provision ratio in order to get comparable performance to the 1:1 Clos.

When coupled with VLB, TROD can no longer outperform Clos. The reason is that, VLB will route many packets via indirect paths, which increases network load and queuing latency in the aggregation layer. The increased queuing latency drastically slows down the FCT.

The traffic-aware approach (Helios) cannot outperform Clos either. The reason is that PoD-level traffic patterns may change within one second. This result indicates that relentlessly pursuing fast reconfiguration may actually hurt performance. Note that, TROD with tVLB reduces FCT by at least 2× when compared with the traffic-aware approach.

The rotation-based approaches also fail to achieve comparable performance with the 1:1 Clos, even if we increase α to 3. The reason is that, without careful coordination between switches and hosts, network congestion slows down FCT. Hence, in order to achieve good FCT for the rotation-based approaches, the switch hardware, the congestion control and flow control schemes, etc., need to be redesigned and co-optimized, which increases the technical barrier.

Finally, the expander graph DCN performs clearly worse than TROD, with FCT 2.4-3.2× higher. The reason is that, expander graphs are optimized for uniform traffic patterns, while practical DCN traffic patterns can be skewed.

5.1.2 Evaluating Synthetic Bursty Traffic Patterns

Although a majority of traffic patterns can be captured by historical traces, unexpected bursts are unavoidable. Since it is hard to find trace segments that cover all the possible burst situations, we create synthetic traces to analyze different DCNs' performance against traffic burst.

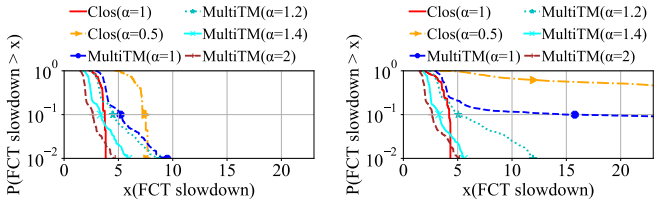
To synthesize busty traffic patterns, we take an arbitrary traffic pattern $D^b = [d^b_{ij}]$ from Facebook's trace as the base, and then add traffic burst on top of D^b. For any $i, j = 1, 2, ..., n$ and $i \neq j$, we create one bursty traffic pattern by increasing d^b_{ij} by certain amount of traffic such that the MLU under the 1:1 Clos reaches a target value, e.g., 0.8. (Note that, a DCN with an MLU of 0.8 is already heavily loaded. Typical data center link utilization is much lower [21], [67].) For every base traffic pattern, we obtain $n^2 - n$ bursty traffic matrices. We repeat this process multiple times, using a different base traffic pattern each time. Then, we evaluate different optical DCNs under these traffic matrices one by one.

The FCT slowdown results are plotted in Fig. 7. Clearly, TROD with the tVLB routing still performs the best, and offers strictly better FCT than Clos when $\alpha = 1.4$. Remind that TROD with ECMP routing performs the second best for the common cases. However, in the bursty cases, ECMP routing may incur severe link congestion, causing many flows unable to finish. TROD with VLB routing performs poorly. Since the MLU of these traces under 1:1 Clos is 0.8, the VLB routing requires $\alpha > 1.6$. However, even with $\alpha = 2$, TROD with VLB still performs worse than 1:1 Clos.

Other than TROD, the traffic-aware approach performs the worst, which cannot finish many flows even with $\alpha = 2$. The performance of the expander graph DCN also deteriorates. The expander graph DCN merely relies on routing to handle the skewed traffic patterns. As network load increases, this approach becomes less effective. Note that when $\alpha < 1.4$, the expander graph DCN experiences severe congestion, dramatically increasing the tail FCT. The Rotation-based approach achieves similar performance under common and bursty traffic patterns, owing to the fact that it is traffic agnostic. However, the Rotation-based approach performs poorly when $\alpha < 2$.

5.2 TROD vs. other Traffic-semi-aware DCNs

5.2.1 Compare with Multi-traffic Designs



(a) Under common traffic. (b) Under synthetic bursty traffic.

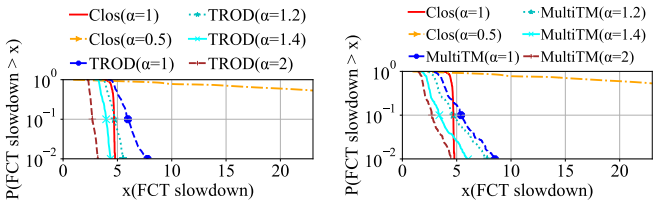
Fig. 8. Performance of MultiTM designs.

Traffic-semi-aware designs [22], [23], [24] are traffic-aware designs with lower control complexity due to the reduced reconfiguration frequency. They use multiple historical TMs (MultiTM) to optimize the topology/routing solution to adapt to the long-term traffic patterns. We fine-tune the number of TMs and the routing weight parameters, and present the best results in Fig. 8, Fig. 9(b) and Fig. 10(b). First, we evaluate the performance under common traffic patterns described in §5.1.1 and synthetic bursty traffic patterns described in §5.1.2. Compared to the traffic-aware design in Fig. 6(d) and Fig. 7(d), the MultiTM design achieves better performance under both common traffic and synthetic bursty traffic, and such improvement is more prominent in the bursty scenario. Compared to TROD in Fig. 6(a) and Fig. 7(a), the MultiTM design degrades FCT performance by about 2×. The reason is that the MultiTM incurs more bandwidth tax in order to handle traffic bursts. To further verify the performance benefit of TROD over the state-of-the-art MultiTM design, we add new comparison results under additional traffic datasets in §5.2.2 and §5.2.3.

5.2.2 Evaluation under High-burst Real Traffic

To improve the fidelity of our evaluation, we search additional DCN traffic datasets for evaluation. [68] analyzes the complexity of different trace sets, and finds that the high-burst pFabric trace is quite different from Facebook's DataBase, WEB and Hadoop datasets (refer to Figure 2 from [68]). Hence, we use the pFabric trace in this section.

The pFabric [69] interconnects 144 servers through 9 leaf switches connected to 4 spine switches in a full mesh. Each leaf switch has 16 10Gbps downlinks (to the server)

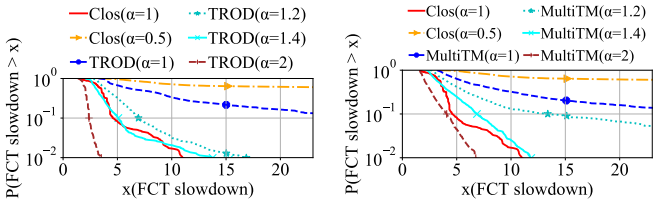


(a) Performance of TROD. (b) Performance of MultiTM.

Fig. 9. Evaluation under pFabric traffic.

and 4 40Gbps uplinks (to the spine) resulting in a non-oversubscribed (full bisection bandwidth) fabric. We aggregate its server-to-server traffic traces to get the approximated leaf-to-leaf traffic traces. To test TROD's performance, we simply treat each leaf as a PoD. pFabric architecture has no aggregation-layer switches and thus the pFabric trace has more multi-position bursts. Fig. 9 shows that TROD exhibits about 1.17×, 1.44×, 1.31×, 1.47× improvements over MultiTM with $\alpha = 1, 1.2, 1.4, 2$ respectively.

5.2.3 *Evaluation under Many-burst Synthetic Traffic*



(a) Performance of TROD. (b) Performance of MultiTM.

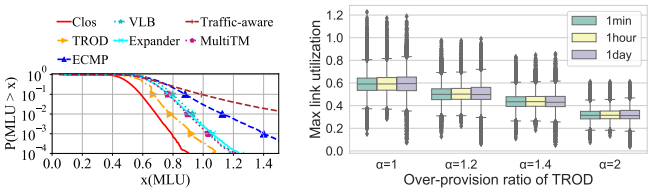
Fig. 10. Evaluation under many-burst traffic.

The experiments in §5.1.2 assume only a single pair of PoDs bursts at any given time. Here, we consider the many-burst case. For every traffic pattern, we add random bursts at $n^2 - n$ positions (the other n positions $d_{ii}, i = 1, 2, \ldots, n$ are intra-PoD traffic demands and we only focus on inter-PoD traffic demands) while guaranteeing that the workload does not exceed the network capacity. Fig. 10 shows that TROD outperforms MultiTM by about 1.15×, 2.16×, 1.36×, 1.69× with $\alpha = 1, 1.2, 1.4, 2$ respectively. The MultiTM designs performs worse than TROD because more traffic is forced to go non-shortest paths. The many-burst cases are not necessarily more difficult to handle than the single-burst cases, because the similarity of traffic patterns may not be low when multiple PoD pairs burst simultaneously.

5.3 Evaluating MLUs Over the Entire Trace

In this experiment, we fix the over-provision ratio of the optical DCNs as 1. We plot the MLU overflow probabilities, e.g., P(MLU>x), in Fig. 11(a). We do not plot the Rotation-based approach because the Rotation-based approach is essentially mesh+VLB if we average its topology over time.

Compared to other optical DCNs and Expander graph DCN, TROD (with tVLB) achieves the best MLU performance. Specifically, if we fix a certain MLU threshold value, e.g., 0.8, TROD's MLU overflow probability is 10× lower than that of the second-best option.



(a) Comparing MLU overflow probabilities. (b) MLU comparison of TROD at different reconfiguration frequency.

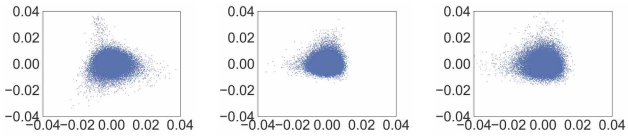
Fig. 11. MLU performance evaluation.

5.4 Dealing with Out-of-order Delivery

As described in §4.2, tVLB routing may cause out-of-order delivery. We recommend enabling selective ack (SACK), to avoid retransmitting packets that arrived out-of-order. Indeed, DCTCP [62] and Swift [70] enables SACK in data centers by default. On the other hand, we recommend using a TCP that does not react to duplicated ACKs (DACK) e.g., DCTCP [62], TCP BBR [71], Swift [70], etc. The reason is that packet reordering may not indicate a packet loss or network congestion under tVLB. If TCP endpoints reduce the congestion window upon receiving three DACKs, the network throughput and the flow completion time would suffer. The following experiment confirms the above analysis.

We randomly select eight sets of TMs from Facebook's trace and the synthetic bursty traffic trace. Every set includes 100 TMs. We simulate TROD with tVLB using DCTCP. We have modified DCTCP so that it can enable/disable reaction to DACKs. The results in Tab. 1 show the 90-percentile and 99-percentile FCT slowdown for each traffic set. Clearly, by disabling the DACK mechanism, TROD achieves better FCT performance, and the performance gap becomes larger as network load increases (see sets 1, 2 and 3).

5.5 TROD's Reconfiguration Frequency



(a) 00:00-07:59 (b) 08:00-15:59 (c) 16:00-23:59

Fig. 12. Visualizing traffic clusters of Facebook's trace using FastICA.

The previous evaluations have adopted daily reconfiguration for TROD. Next, we evaluate TROD's performance over different reconfiguration frequencies. We compare the MLU performance under daily, hourly and minutely reconfigurations in Fig. 11(b) using Facebook's traces. Surprisingly, daily reconfiguration achieves similar MLU performance, when compared with the other two options. To understand the reason, we perform FastICA [72] for Facebook's one-day trace to visualize the trace's traffic clusters. Fig. 12 suggests that there is only one traffic cluster, and this traffic cluster does not change much with respect to time.

TABLE 1
FCT slowdown with and without duplicated ACK (DACK).

	No DACK 90%	No DACK 99%	DACK 90%	DACK 99%
Set 1	6.5	13.6	11.2	31.2
Set 2	4.5	7.1	6.5	12.2
Set 3	3.4	4.4	5.3	7.8
Set 4	2.4	3.3	3.8	5.7
Set 5	4.7	4.8	4.7	5.6
Set 6	3.6	3.7	4.6	4.7
Set 7	3.0	3.1	3.6	3.8
Set 8	1.9	2.0	2.5	2.6

The observation in Fig. 12 applies to all the three DCN clusters. Admittedly, Facebook's trace may not be representative for all the data centers. There may be data centers that have different application mix during different times of a day, resulting in multiple distinct traffic clusters. Nonetheless, the lesson is, faster reconfiguration is not always better.

The above findings based on Facebook's traces coincide with Google's findings. In Google's Jupiter, traffic patterns change over time, but PoD-level reconfiguration more frequent than every few weeks yields limited benefits [23], [24].

5.6 Sensitivity Analysis

Alg. 1 shows that TROD requires a percentile value p as an input parameter. Here, we study the sensitivity of TROD to different p values. In the section, average hop count (AHC) is also an important metric, which is equal to the total traffic that traverses h hop divided by the total traffic demand.

For different values of p ranging from 0.5 to 0.95, we compute different topology+routing solutions using TROD, and evaluate all the TMs to obtain average or different percentile values of MLU and AHC. The results are summarized in Table 2 and Table 3. As expected, the AHC values increases slightly as p decreases. This is because the routing thresholds decrease, causing more traffic being routed to two-hop paths. The MLU values are more interesting in database cluster. As we decrease p, the lower percentile values (up to 95th percentile) of MLU decrease, while the higher percentile values (99th percentile) increase. The reason is that, when only a few traffic demands burst, which corresponds to the majority cases, load balancing more traffic to two-hop paths reduces MLU. However, when many traffic demands burst at the same time, which corresponds to the worst cases, load balancing more traffic to two-hop paths may actually increase the overall network load, causing a larger MLU. Fortunately, the MLU values do not change significantly as we change p.

Note that the web search cluster is different from the other two clusters in terms of the MLU values, i.e., the MLU values always decrease as we decrease p. This indicates that the web search cluster is less bursty compared to the other two clusters. For the Hadoop cluster, the MLU values increase as p decreases starting from 95th percentile, which is earlier than that of the database cluster. This indicates that the Hadoop cluster is more bursty than the database cluster.

5.7 Micro-benchmark of tVLB on P4

In this section, we first evaluate the rate control accuracy for splitting traffic under given thresholds, then we benchmark the hardware resource consumption for implementing tVLB in data plane.

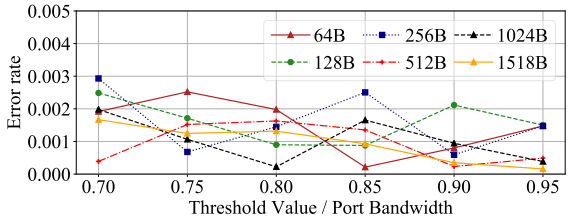


Fig. 13. Error rate of splitting traffic.

For the accuracy test, we configured a Wedge100BF-32X switch to meter the ingress traffic and forward the part that exceeds the threshold to a different output port from the original one. We use the Cisco TRex packet generator to feed a constant bitrate packet stream to the switch through an Intel 82599 10GbE NIC, and the throughput going out of different ports is measured by the switch's port manager. We varied the threshold value from 7Gbps to 9.5Gbps as well as the packet size in the constant bitrate stream from 64Bytes to 1518Bytes, and the error rate is calculated as the ratio of the actual throughput bias to the desired one. Fig. 13 shows that error rate of splitting traffic are negligible (<0.3%) under all cases we have tested. The high accuracy comes from the hardware-implemented meter in our P4 switch design.

We also measured the hardware resource usage of tVLB on Tofino switches, and results are normalized with the values of switch.p4 because the Non-Disclosure Agreement limits us to showing the absolute values. The results in Table 4 show that our tVLB consumes only a small number of resources which means it leaves abundant resources for other data plane network functions. Worth noting that switch.p4 is designed for stateless packet forwarding, so it consumes a small count of RAMs and Meter ALUs, which makes the normalized usage seem large. In fact, tVLB only consumes a small portion of the overall resources.

5.8 Cost Analysis

We analyze the equipment and energy cost for TROD with $\alpha = \alpha_T$, the 1:1 Clos, and the Rotation-based optical DCN with $\alpha = \alpha_R$. Consider a DCN with m servers. Within a PoD, copper cables are used to connect servers to ToRs and connect ToRs to Aggregation switches. Since different PoDs and the core layer switches may be located tens of meters apart, fibers are used for interconnection. Table 5 summarizes the main sources of deployment cost for different components of a DCN. The 1:1 Clos requires $5m$ electrical ports, $2m$ optical transceivers, m fibers and $2m$ copper cables. The TROD with over-provision ratio α_T requires $(3 + \alpha_T)m$ electrical ports, $\alpha_T m$ OCS ports, $\alpha_T m$ optical transceivers, $\alpha_T m$ fibers and $2m$ copper cables. The Rotation-based approach with over-provision ratio α_R requires $(1 + \alpha_R)m$ electrical ports, $\alpha_R m$ OCS ports, $\alpha_R m$ optical transceivers, $\alpha_R m$ fibers and m copper cables. Table 5 also lists the energy cost. In practice, the energy consumption of a 320-port OCS is less than 45 Watts [25], which is negligible when compared to the energy consumption of electrical switches.

The exact prices of different network devices vary with many factors. Here, We provide the relative quantity rela-

TABLE 2
MLU under different TROD p parameter in different cluster.

Database						Web Search						Hadoop					
p	avg	80th	90th	95th	99th	p	avg	80th	90th	95th	99th	p	avg	80th	90th	95th	99th
0.95	0.492	0.568	0.598	0.624	0.638	0.95	0.637	0.735	0.759	0.804	0.819	0.95	0.507	0.544	0.558	0.58	0.649
0.9	0.487	0.552	0.588	0.588	0.637	0.9	0.634	0.72	0.766	0.774	0.787	0.9	0.493	0.518	0.537	0.567	0.652
0.8	0.482	0.543	0.555	0.58	0.644	0.8	0.633	0.724	0.726	0.739	0.765	0.8	0.465	0.487	0.52	0.567	0.68
0.7	0.48	0.528	0.554	0.58	0.649	0.7	0.631	0.697	0.703	0.719	0.75	0.7	0.456	0.482	0.521	0.576	0.698
0.6	0.479	0.527	0.556	0.583	0.656	0.6	0.611	0.655	0.676	0.695	0.734	0.6	0.449	0.483	0.531	0.593	0.721
0.5	0.48	0.528	0.558	0.587	0.664	0.5	0.602	0.644	0.67	0.692	0.734	0.5	0.448	0.49	0.543	0.609	0.74

TABLE 3
AHC under different TROD p parameter in different cluster.

Database						Web Search						Hadoop					
p	avg	80th	90th	95th	99th	p	avg	80th	90th	95th	99th	p	avg	80th	90th	95th	99th
0.95	1.005	1	1.014	1.036	1.086	0.95	1.001	1	1.004	1.008	1.017	0.95	1.005	1.007	1.014	1.024	1.057
0.9	1.008	1.01	1.03	1.053	1.103	0.9	1.003	1.004	1.009	1.014	1.026	0.9	1.009	1.014	1.024	1.04	1.084
0.8	1.018	1.033	1.056	1.079	1.132	0.8	1.006	1.012	1.019	1.026	1.04	0.8	1.028	1.04	1.062	1.093	1.158
0.7	1.025	1.046	1.07	1.094	1.148	0.7	1.011	1.02	1.03	1.039	1.056	0.7	1.042	1.061	1.088	1.125	1.196
0.6	1.032	1.058	1.083	1.108	1.161	0.6	1.02	1.035	1.049	1.061	1.083	0.6	1.065	1.092	1.127	1.17	1.246
0.5	1.043	1.073	1.099	1.124	1.178	0.5	1.027	1.047	1.063	1.076	1.1	0.5	1.094	1.13	1.169	1.217	1.297

TABLE 4
tVLB hardware resource consumption normalized with switch.p4.

Resource Category	Match Crossbar	Hash Bits	Meter ALU	Stats ALU	Map RAM
Normalized Usage	0.37%	0.73%	14.3%	0	6.7%
Resource Category	SRAM	VLIW	TCAM	Gateway	
Normalized Usage	1.1%	0.70%	0	0	

tionship formula (see Table 5), which the readers can use to make what they think is a reasonable estimate. Summing up the cost of different network components, the total cost of the 1:1 Clos is approximately $1890m$ and the total cost of TROD is approximately $(775 + 710\alpha_T)m$. Hence, TROD will be cheaper than Clos as long as $\alpha_T < 1.57$. Note that TROD with $\alpha_T = 1.2$ achieves comparable performance as Clos, and TROD with $\alpha_T = 1.4$ performs strictly better.

The total cost of Rotation-based approach is approximately $(260 + 700\alpha_R)m$. Rotation-based approach with $\alpha_R = 2$ is cost comparable to TROD with $\alpha_T = 1.2$; Rotation-based approach with $\alpha_R = 2.5$ costs more than TROD with $\alpha_T = 1.4$. However, Rotation-based approach achieves about 1.5-2$\times$ longer FCT (see Fig. 6(e) & 7(e)).

TABLE 5
Cost Analysis for DCNs with m Servers.

Components	1:1 Clos	TROD (α_T)	Rotation (α_R)
Equipment Cost			
ToR layer	$2mP_e$	$2mP_e$	$(1+\alpha_R)mP_e$
Aggr layer	$2mP_e$	$(1+\alpha_T)mP_e$	0
Transceiver	$2mP_t$	$\alpha_T mP_t$	$\alpha_R mP_t$
Copper Cable	$2mP_c$	$2mP_c$	mP_c
Fiber	mP_f	$\alpha_T mP_f$	$\alpha_R mP_f$
Core Layer	mP_e	$\alpha_T mP_g$	$\alpha_R mP_r$
Energy Cost			
Switch energy	$5mE_s$	$(3+\alpha_T)mE_s$	$(1+\alpha_R)mE_s$
OCS energy	0	$\alpha_T mE_o$	$\alpha_R mE_o$

6 Discussion on DCN Complexity

Complexity is always an important consideration for data centers. As network demand grows gradually, data centers may require incremental expansion [31], which is related to **deployment complexity**. During incremental expansion, additional capacity is installed first, and then the DCN topology needs to be reconfigured. Topology reconfiguration is easy for TROD, because all the PoDs are interconnected by a layer of OCSs. Without OCSs, topology reconfiguration would require significant amount of labor work.

For **hardware complexity**, [12] and [13] build an OCS prototype with microsecond-level reconfiguration delay using wavelength selective switching. However, this OCS prototype has limited port count, and thus using this prototype to build optical DCNs hurts network scalability. Some other works [14], [15], [73] use steerable wireless transceivers to improve scalability. However, their sophisticated steering mechanisms and wireless solutions typically face other deployment challenges, e.g., they are sensitive to environmental conditions in real DCNs. [17], [18] and [6] uses rotor switches or AWGRs to form a uniform mesh by rotating the network topology among several topology configurations. However, this approach requires even lower reconfiguration delay and an extremely high-accuracy synchronization protocol, and thus is still early for mass production. In contrast, TROD does not need fast optical reconfiguration and only uses commercial OCSs [25] to build optical DCNs.

For **control complexity**, the pioneer works Helios [10] and c-through [11] need to distinguish elephant and mice flows, but there is no accuracy guarantee. Many subsequent works [6], [17], [18] require high-frequency reconfigurations, but the convergence speed of the state-of-art SDN controller is too slow. Chopin [74] uses a threshold to balance the control complexity and performance between centralized and distributed control. TROD does not require accurate traffic prediction and only conducts low-frequency topology/routing reconfigurations with commercial OCSs, dra-

matically reducing the control complexity of optical DCNs.

For **algorithmic complexity**, COUDER [22] or Google's Jupiter DCN [23] [24] need to solve three linear programming (LP) problems to find a good topology solution, and each LP problem could contain nearly a million decision variables. In contrast, TROD does not need accurate traffic prediction or solving LP problems, and its algorithmic complexity is much lower (refer to §3.3).

7 Conclusion

We proposed TROD, a threshold-based routing-topology co-design using off-the-shelf OCS with no need for high-frequency reconfiguration, that achieves better FCT than the existing optical DCNs and the expander graph DCNs. With capacity over-provision at the OCS layer, TROD may even outperform the non-oversubscribed Clos. Compared with other optical DCNs, TROD has low deployment complexity, owing to the fact that it does not require customized switch hardware and host modification; TROD also has low management complexity, due to the fact that it does not need to react to every traffic change.

Definitely, TROD is not the eventual architecture of the optical DCN. As link speed increases, the power cost of multi-layer DCN architectures may become prohibitive. Then, a potential next step of TROD is to study, if it is possible to extend the design principles of TROD to the ToR layer interconnect. The challenge is that, 1) the ToR-layer DCN traffic exhibits much higher uncertainty and 2) a ToR switch has much fewer uplinks and the ToR-level logical topology is much more sparse. One possible solution is to construct a few groups of ToRs, and then study how to design a topology among ToR groups. With TROD, we hope network vendors can be convinced to deploy optical DCNs, and accelerate the evolution towards the eventual goal of full optical DCN in the future.

Acknowledgments

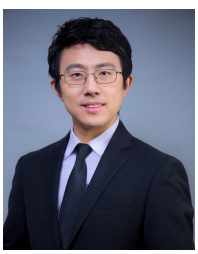
Thank our editors and reviewers. This work was supported by the NSF China (No. 42050105, 62272292 and 61902246).

References

[1] P. Cao, S. Zhao, M. Y. Teh, Y. Liu, and X. Wang, "Trod: Evolving from electrical data center to optical data center," in *ICNP*, 2021.
[2] Intel, "Affordably increase network bandwidth at 100 gbps and beyond," https://rb.gy/2h1ldp, 2020.
[3] Sylex, "Where are the il limits for 10 40 and 100gbps applications," https://rb.gy/8kkydz.
[4] Juniper, "Migrating to a 40 gbps data center," https://www.juniper.net/us/en/local/pdf/whitepapers/2000578-en.pdf, 2015.
[5] H. Yu, P. Doussiere *et al.*, "400gbps fully integrated dr4 silicon photonics transmitter for data center applications," in *OFC*, 2020.
[6] H. Ballani, P. Costa, R. Behrendt *et al.*, "Sirius: A flat datacenter network with nanosecond optical switching," in *SIGCOMM*, 2020.
[7] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM*, 2008.
[8] A. Singh, J. Ong, A. Agarwal, G. Anderson *et al.*, "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," in *SIGCOMM*, 2015.
[9] A. Greenberg, J. R. Hamilton *et al.*, "Vl2: A scalable and flexible data center network," in *SIGCOMM*, 2009.
[10] N. Farrington, G. Porter *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," in *SIGCOMM*, 2010.
[11] G. Wang, D. G. Andersen, M. Kaminsky *et al.*, "c-through: Part-time optics in data centers," in *SIGCOMM*, 2010.
[12] G. Porter, R. Strong, N. Farrington *et al.*, "Integrating microsecond circuit switching into the data center," in *SIGCOMM*, 2013.
[13] H. Liu, F. Lu, A. Forencich *et al.*, "Circuit switching under the radar with reactor," in *NSDI*, 2014.
[14] N. Hamedazimi, Z. Qazi *et al.*, "Firefly: A reconfigurable wireless data center fabric using free-space optics," in *SIGCOMM*, 2014.
[15] M. Ghobadi, R. Mahajan, A. Phanishayee *et al.*, "Projector: Agile reconfigurable data center interconnect," in *SIGCOMM*, 2016.
[16] L. Chen, K. Chen, Z. Zhu *et al.*, "Enabling wide-spread communications on optical fabric with megaswitch," in *NSDI*, 2017.
[17] W. M. Mellette, R. Mcguinness, A. Roy, A. Forencich, and G. Porter, "Rotornet: A scalable, low-complexity, optical datacenter network," in *SIGCOMM*, 2017.
[18] W. M. Mellette, R. Das, Y. Guo *et al.*, "Expanding across time to deliver bandwidth efficiency and low latency," in *NSDI*, 2020.
[19] R. Zhang-Shen and N. McKeown, "Designing a fault-tolerant network using valiant load-balancing," in *INFOCOM*, 2008.
[20] C. Delimitrou *et al.*, "Echo: Recreating network traffic maps for datacenters with tens of thousands of servers," in *IISWC*, 2012.
[21] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *SIGCOMM*, 2015.
[22] M. Y. Teh, S. Zhao, P. Cao *et al.*, "Enabling quasi-static reconfigurable networks with robust topology engineering," in *ToN*, 2022.
[23] L. B. Poutievski, O. Mashayekhi, J. S. Ong *et al.*, "Jupiter evolving: transforming google's datacenter network via optical circuit switches and software-defined networking," in *SIGCOMM*, 2022.
[24] M. Zhang, J. Zhang, R. Wang *et al.*, "Gemini: Practical reconfigurable datacenter networks with topology and traffic engineering," in *ArXiv*, vol. abs/2110.08374, 2021.
[25] I. CALIENT Technologies, https://www.calient.net/.
[26] I. Agiltron, https://agiltron.com/.
[27] I. Polatis, https://www.polatis.com/.
[28] Y. Geng, S. Liu, Z. Yin *et al.*, "Exploiting a natural network effect for scalable, fine-grained clock synchronization," in *NSDI*, 2018.
[29] M. Y. Teh, Y.-H. Hung *et al.*, "Tago: Rethinking routing design in high performance reconfigurable networks," in *SC*, 2020.
[30] Y. Li, R. Miao, H. Liu, Y. Zhuang, F. Feng *et al.*, "Hpcc: High precision congestion control," in *SIGCOMM*, 2019.
[31] S. Zhao, R. Wang *et al.*, "Minimal rewiring: Efficient live expansion for clos data center networks: Extended version," in *NSDI*, 2019.
[32] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: fine grained traffic engineering for data centers," in *CoNEXT*, 2011.
[33] M. Al-Fares, S. Radhakrishnan *et al.*, "Hedera: Dynamic flow scheduling for data center networks," in *NSDI*, 2010.
[34] D. Szostak, A. Włodarczyk, and K. Walkowiak, "Machine learning classification and regression approaches for optical network traffic prediction," in *Electronics*, 2021.
[35] M. Wang, Y. Cui *et al.*, "Neural network meets dcn: Traffic-driven topology adaptation with deep learning," in *SIGMETRICS*, 2018.
[36] F. Morales, M. Ruiz, and L. Velasco, "Data analytics based origin-destination core traffic modelling," in *ICTON*, 2017.
[37] C. Streiffer *et al.*, "Deepconf: Automating data center network topologies management with machine learning," in *NetAI*, 2018.
[38] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," in *J. Light. Technol.*, 2018.
[39] M. Balanici and S. Pachnicke, "Machine learning-based traffic prediction for optical switching resource allocation in hybrid intra-data center networks," in *OFC*, 2019.
[40] T. T-H *et al.*, "Virtual network embedding in ring optical data centers using markov chain probability model," in *TNSM*, 2019.
[41] T. Truong-Huu *et al.*, "Markov chain based algorithm for virtual network embedding in optical data centers," in *HPCC*, 2016.
[42] L. Luo *et al.*, "Optimizing multicast flows in high-bandwidth reconfigurable datacenter networks," in *J. Netw. Comput. Appl.*, 2022.
[43] L. Luo, K.-T. Foerster *et al.*, "Splitcast: Optimizing multicast flows in reconfigurable datacenter networks," in *INFOCOM*, 2020.
[44] T. Fenz, K.-T. Foerster *et al.*, "Efficient non-segregated routing for reconfigurable demand-aware networks," in *IFIP*, 2019.
[45] M. N. Hall, K.-T. Foerster, S. Schmid, and R. Durairajan, "A survey of reconfigurable optical networks," in *OSN*, 2021.
[46] A. D. Ferguson, S. D. Gribble *et al.*, "Orion: Google's software-defined networking control plane," in *NSDI*, 2021.
[47] S. Zhao, P. Cao, and X. Wang, "Understanding the performance guarantee of physical topology design for optical circuit switched data centers," in *SIGMETRICS*, 2022.

[48] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," in *Journal of the ACM*, 1988.
[49] K.-T. Foerster *et al.*, "Characterizing the algorithmic complexity of reconfigurable data center architectures," in *ANCS*, 2018.
[50] J. Zhou, M. Tewari *et al.*, "Wcmp: Weighted cost multipathing for improved fairness in data centers," in *EuroSys*, 2014.
[51] O. N. Fundation, "Openflow switch specification 1.3.1," in *ONF Technical Report*, 2012.
[52] P. D. P. at Terabit Speeds, https://rb.gy/fbfqpr, 2018.
[53] E. L. Fernandes, E. Rojas *et al.*, "The road to bofuss: The basic openflow userspace software switch," in *JNCA*, 2020.
[54] F. Chen, C. Wu, X. Hong *et al.*, "Engineering traffic uncertainty in the openflow data plane," in *INFOCOM*, 2016.
[55] L. J. Chaves *et al.*, "Ofswitch13: Enhancing ns-3 with openflow 1.3 support," in *Proceedings of the Workshop on ns-3*, 2016.
[56] K. Tantayakul, R. Dhaou *et al.*, "Experimental analysis in sdn open source environment," in *ECTI-CON*, 2017.
[57] V. Šulák, P. Helebrandt, and I. Kotuliak, "Performance analysis of openflow forwarders based on routing granularity in openflow 1.0 and 1.3," in *FRUCT*, 2016.
[58] huawei switch manual, https://rb.gy/kru6vw.
[59] pic8 switch manual, https://rb.gy/aemx4f.
[60] cisco switch manual, https://rb.gy/fy0lks.
[61] TROD, https://github.com/caopeirui/TROD.
[62] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye *et al.*, "Data center tcp (dctcp)," in *SIGCOMM*, 2010.
[63] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," in *SIGCOMM Review*, 2006.
[64] Netbench, https://github.com/ndal-eth/netbench.
[65] A. Valadarsky, M. Dinitz, and M. Schapira, "Xpander: Unveiling the secrets of high-performance datacenters," in *HotNets*, 2015.
[66] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, "Xpander: Towards optimal-performance datacenters," in *CoNEXT*, 2016.
[67] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *SIGCOMM*, 2010.
[68] C. Avin, M. Ghobadi, C. Griner, and S. Schmid, "On the complexity of traffic traces and implications," in *SIGMETRICS*, 2020.
[69] M. Alizadeh, S. Yang *et al.*, "pfabric: minimal near-optimal datacenter transport," in *SIGCOMM*, 2013.
[70] G. Kumar, N. Dukkipati *et al.*, "Swift: Delay is simple and effective for congestion control in the datacenter," in *SIGCOMM*, 2020.
[71] N. Cardwell, Y. Cheng *et al.*, "Bbr: Congestion-based congestion control," in *Communications of the ACM*, 2017.
[72] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," in *Neural Networks*, 2000.
[73] X. Zhou, Z. Zhang, Y. Zhu, Y. Li *et al.*, "Mirror mirror on the ceiling: flexible wireless links for data centers," in *SIGCOMM*, 2012.
[74] N. Rozen-Schiff, K.-T. Foerster, S. Schmid, and D. Hay, "Chopin: Combining distributed and centralized schedulers for self-adjusting datacenter networks," in *OPODIS*, 2023.

Peirui Cao received the B.S. degree from Southeast University and the M.S. degree from Beihang University. He is currently pursuing the Ph.D. degree with the John Hopcroft Center for Computer Science, Shanghai Jiao Tong University. He has published papers in top-tier conferences and journals, including NSDI, SIGMETRICS, ICNP, IEEE/ACM TRANSACTIONS ON NETWORKING. His research interests include optimizing network performance, data center networks, and network measurement.



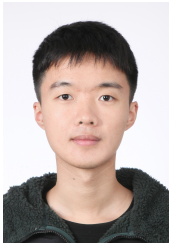
Shizhen Zhao received the bachelor's degree from Shanghai Jiao Tong University in 2010 and the Ph.D. degree from Purdue University in 2015. He is currently a Tenure-Track Associate Professor with the John Hopcroft Center, Shanghai Jiao Tong University. Before joining SJTU, he worked in Google's Networking Team, managing Google's hyper-scale data center networks. He has published papers in top-tier conferences and journals, including NSDI, SIGMETRICS, MOBICOM, ICNP, INFOCOM, IEEE/ACM ToN, and IEEE TAC. His current research interest is optimizing optical circuit-switched data center networks.

Dai Zhang received the B.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2015. He is currently pursuing the Ph.D. degree with the Institute for Network Sciences and Cyberspace, Tsinghua University. His research interests include software-defined networking and programmable data planes.

Zhuotao Liu received a Ph.D. from University of Illinois at Urbana-Champaign, USA. He is currently an assistant professor of Institute for Network Sciences and Cyberspace, Tsinghua University. Before joining Tsinghua, he was a technical lead at Google, managing massive-scale software-defined datacenter networks. His research interests include data security & privacy, blockchain & applied cryptography, and secure Internet architecture.

Mingwei Xu is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua. His research interests include future Internet architecture, Internet routing, and network virtualization. He won the second prize of national scientific and technological progress award for three times and the second prize of national technology invention award. He is the winner of National Science Foundation for Distinguished Young Scholars of China. He has served as the TPC chair or a member for several IEEE conferences, such as ICPP, Globecom, and ICC.

Min Yee Teh received his B.S. degree in 2016 from Brown University, and his M.S. degree in 2017 and Ph.D. degree in 2021 from Columbia University, all in Electrical Engineering. His Ph.D. work was supported by the Wei Family Foundation fellowship. Teh currently works in Google's network infrastructure team as a software engineer. His research interests lie in reconfigurable optical circuit-switched networks, load balancing, and data center network topology design.



Yunzhuo Liu received his B.S. degree from Shanghai Jiao Tong University and is now a PhD candidate in John Hopcroft Center at Shanghai Jiao Tong University. He has published papers in top-tier conferences, including SIGMETRICS, INFOCOM, ACM MM and ICNP. His research interests include distributed training and programmable networks.

Xinbing Wang received the B.S. degree (with hons.) from Shanghai Jiao Tong University in 1998, and the M.S. degree from Tsinghua University in 2001. He received the Ph.D. degree from North Carolina State University in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiao Tong University. Dr. Wang has been an associate editor for ToN, TMC, and the member of the Technical Program Committees of several conferences including MobiCom, MobiHoc, INFOCOM.

Chenghu Zhou received the B.S. degree from Nanjing University in 1984, and the M.S. degree and Ph.D. degree from Chinese Academy of Sciences. He is an academician of Chinese Academy of Sciences and an academician of International Eurasian Academy of Sciences. Currently, he is the director of the State Key Laboratory of Resources and Environmental Information System. His research of interests are in the area of spatial data mining, geographic system modeling, hydrology and water resources, geographic information system and remote sensing application.