TROD: Evolving From Electrical Data Center to Optical Data Center

Peirui Cao, Shizhen Zhao*, Min Yee Teh‡, Yunzhuo Liu, Xinbing Wang
Shanghai Jiao Tong University, ‡*Columbia University*
{*caopeirui, shizhenzhao, liu445126256, xwang8*}*@sjtu.edu.cn*,
‡*mt3126@columbia.edu*

2021 IEEE 29th International Conference on Network Protocols (ICNP) | 978-1-6654-4131-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICNP52444.2021.9651977

***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 tremendous deployment complexity. Second, these designs are not full-optical and the performance benefit against the non-blocking Clos DCN is not clear.**

After exploring the design tradeoffs of the existing optical DCN designs, we propose TROD (Threshold Routing based Optical Datacenter), 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 VLB), 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 at least 2× compared with the existing optical DCN designs, and by approximately 2.4-3.2× compared with 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.

I. 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 [1]–[4]. However, electrical switching is becoming cost-and-energy prohibitive to keep up with the bandwidth scaling [5]. This trend has driven the development of Optical Circuit Switches (OCS) to build the future 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 [6]–[8]. 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 [9]–[15] 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 also 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 [16], Opera [17] or Sirius [5], 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** in the rest of this paper), and then use valiant load balancing [18] (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 [5], 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. 4(e) & 5(e) in §IV).

Motivated by Microsoft and Facebook's trace studies [19], [20] that PoD-level DCN traffic has certain recurring patterns, we identify a third opportunity that has never been taken before, and propose TROD, a low-complexity traffic-semi-agnostic PoD-level optical DCN. We perform optical switching based on the long-term recurring patterns instead of time-varying patterns. Since most future traffic patterns can be covered by these recurring patterns, this design does not have to reconfigure OCSs based on the real time demand changes.

Compared with the ToR-level optical DCNs, TROD requires neither customized switch hardwares nor host modifications. Compared with the hybrid electrical/optical DCNs, TROD only maintains one optical core layer and does not need to perform traffic classification between latency-sensitive and latency-tolerant flows. More importantly, TROD does not need to react to every traffic change, which may significantly lower the control and management complexity of optical DCNs.

The key challenge of TROD is to deal with traffic uncertainty without frequent OCS reconfigurations. At the PoD level, although most future traffic patterns can be captured by

* Shizhen Zhao is the corresponding author.

historical traces, we may still encounter unpredictable traffic bursts. Motivated by the traffic-agnostic VLB routing, we proposed a threshold-based VLB (tVLB) routing protocol for TROD, that properly handles such unexpected traffic bursts. 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. Compared with the existing optical DCN designs, TROD reduces FCT by at least 2$\times$. 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.

II. 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 [5]. 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.

A. Hybrid Core vs. Full Optical Core

Since DCN traffic is highly bursty and the commercially available OCSs [21]–[23] have a large reconfiguration delay around 30ms, the initial attempts [9], [10] 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.

B. Traffic Aware vs. Agnostic Designs

Consider optical DCNs with only optical cores. In order to handle latency sensitive traffic, the conventional wisdom [11]–[14] 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 [15]. Although ProjectTor [14] 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. 4(d) & Fig. 5(d)).

Due to the difficulty of handling fine grained traffic changes, [5], [16], [17] proposed a Rotation-based architecture using either rotor switch [16] or AWGR [5]. 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 [5], 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. 4(e) & Fig. 5(e)).

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.

C. 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 [25] and incremental expansion [26] 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 DCN traffic, PoD-level traffic exhibits some spatial patterns [19], [20]. 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 §III-F), 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 [5]. Indeed, except for the pioneer works [9] that adopt a PoD-level design, other follow-up works [5], [10]–[17] 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 [27]. 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 [5], [16], [17], 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 [10], [12].

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.

D. Is High-Frequency Reconfiguration Necessary?

Clearly, the right reconfiguration frequency depends on the traffic stability characteristics. We perform a trace analysis using Facebook's public trace [20] 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 TMs[2]. Fig. 1 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 [9] proposed using the currently-seen TM to reconfigure topology. Due to the instability of DCN traffic, this approach yields poor network performance (see our simulation results in Fig. 4(d) & Fig. 5(d)).

Second, PoD-level DCN traffic does have a weaker form of stability, i.e., although the traffic pattern changes all the time,

[1]Note that the most recent literature on data center scale time synchronization could only achieve an accuracy of tens of nanoseconds [24].

[2]Here we view each traffic matrix (TM) as a vector.

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 also plot such similarity values in Fig. 1. 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.

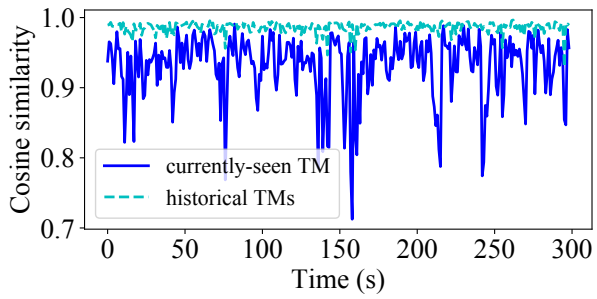


Fig. 1. Cosine similarity analysis using the 5-minute historical TMs vs. using the currently-seen TM.

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. 6 & 4(a) & 5(a)). In contrast, frequent reconfiguration based on the currently-seen traffic pattern even hurts performance (Fig. 4(d) & 5(d)).

III. 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. In addition, we show how to implement tVLB and reconfigure OCSs and switches.

A. TROD's Physical Structure

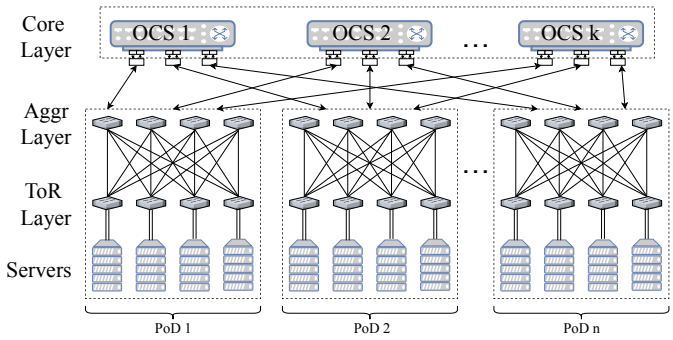


Fig. 2. Datacenter structure of TROD

After exploring the design tradeoffs of the existing optical DCN proposals, we propose TROD, a high-performance optical DCN with low deployment complexity. The network architecture is shown in Fig. 2. The DCN 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-fly 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 [9] and TROD perform ToE in the PoD layer, however, 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 (§III-B) and topology (§III-C) 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 §IV 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).

Mathematical Notations: For ease of reference, notations are summarized in Table 1.

B. 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:

TABLE I
NOTATIONS USED IN THIS PAPER

n	Total number of PoDs.
$D(t) = [d_{ij}(t)]$	Traffic matrices, where $d_{ij}(t)$ is the aggregated data rate (Gbps) of all flows from PoD i to PoD j.
$X = [x_{ij}]$	Inter-Pod topology, where x_{ij} is the number of connections established by the OCS layer between the transmitters of PoD i and the receivers of PoD j.
r_i	The number of bidirectional links between PoD i and the OCS layer.
B	The per-port capacity (Gbps).
u_{ij}	Link utilization between PoD i and PoD j.
C_{ij}	Total capacity from PoD i to PoD j, where $C_{ij} = Bx_{ij}$.
s_{ij}	Final threshold value of the demand d_{ij}.
$s_{ij}(p)$	Initial threshold value of the demand d_{ij} as the p-th percentile value of $d_{ij}(t)$'s.
$\gamma_{i(k)j}$	The fraction of the excessive traffic $(d_{ij} - s_{ij})^+$ that is routed to the 2-hop path $i \to k \to j$ in tVLB routing.
z^+	$\max\{0, z\}$.
M	Number of aggregation switches in a PoD.
α	Over-provision ratio defined in § IV.

- 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.

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. 4(b) & 5(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. 4(c) & 5(c)). tVLB combines the benefits of both worlds. With a proper set of 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.

C. 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. 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

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

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 detailed algorithm is shown in Alg. 1. 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.

Note that 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 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 through direct-hop paths. These two cases are both accommodated in lines 19-20 of Alg. 1.

D. Performance Guarantee of TROD

To understand the performance of TROD, we characterize an inner bound of TROD's capacity region[3] 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:

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

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

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

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 $S = [s_{ij}]$;

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

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

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

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.

E. Implementing tVLB

TROD uses tVLB for routing. tVLB can be easily supported with the commercially available programmable electrical switches, including switches that support OpenFlow 1.3 [28] or P4 [29].

We briefly discuss the implementation using OpenFlow 1.3 switches. Openflow 1.3 switches support two important features: multiple flow table (MFT) pipeline and meter table [28], which can be used to implement tVLB as follows:

Step 1: Define a meter with band type as "dscp remark", and use the desired threshold as the rate limit.

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, 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; while packets with modified DSCP values are forwarded to the set of indirect paths.

We have tested the above design in *ofsoftswitch13* [30], and thus any hardware switch that fully supports OpenFlow 1.3 [31]–[34] should be able to implement tVLB.

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 solve the problem with existing protocols. More details are available in §IV-B.

F. Reconfiguring OCSs and Switches

TROD's routing and topology solutions are designed using a sequence of historical traffic matrices 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 too much capacity during reconfiguration, especially when the network load is high, TROD may perform OCS configurations in multiple steps. To reduce the number of reconfiguration steps, TROD adopts minimal rewiring [26] 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.

IV. 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.

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 [35]. Since FCT is hard to compute mathematically, we instead perform packet-level simulation under different DCNs using our packet-level simulator[4], which is extended from an open source network simulator NetBench [36]. Note that Facebook's traces contain flows of different sizes. To allow better comparison across flows and DCNs, we use ***FCT slowdown*** [25], 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. 3 for different DCN architectures.

[4]Our packet-level simulator is available at https://github.com/caopeirui/TROD

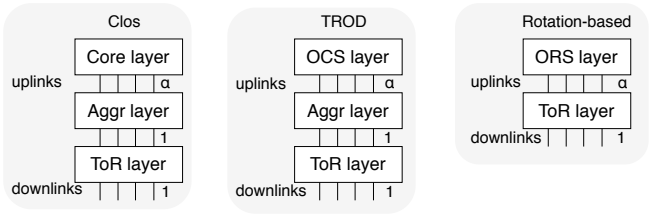


Fig. 3. Illustrating over-provision ratio α over different DCN architectures.

A. TROD vs. Other DCNs

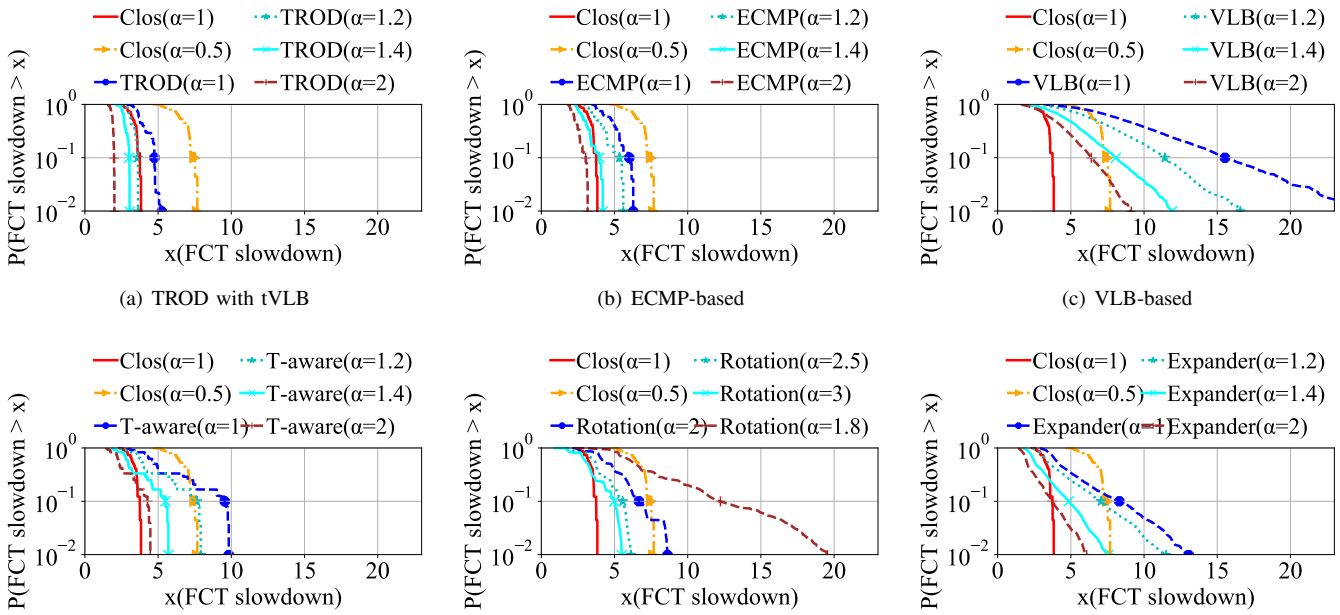
We evaluate FCT slowdown using Facebook's production traces [20] for the following DCN architectures:

1) 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.
2) Traffic-aware optical DCN (Helios [9]): 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.
3) Traffic-agnostic Rotation-based optical DCN [5], [16], [17]: Rotate OCS configurations every 100ns, and then use VLB for routing. [5] 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.)
4) Expander graph DCN [37], [38]: 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.

1) Evaluating Common Traffic Patterns: NetBench is a discrete event simulator. 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. 4.

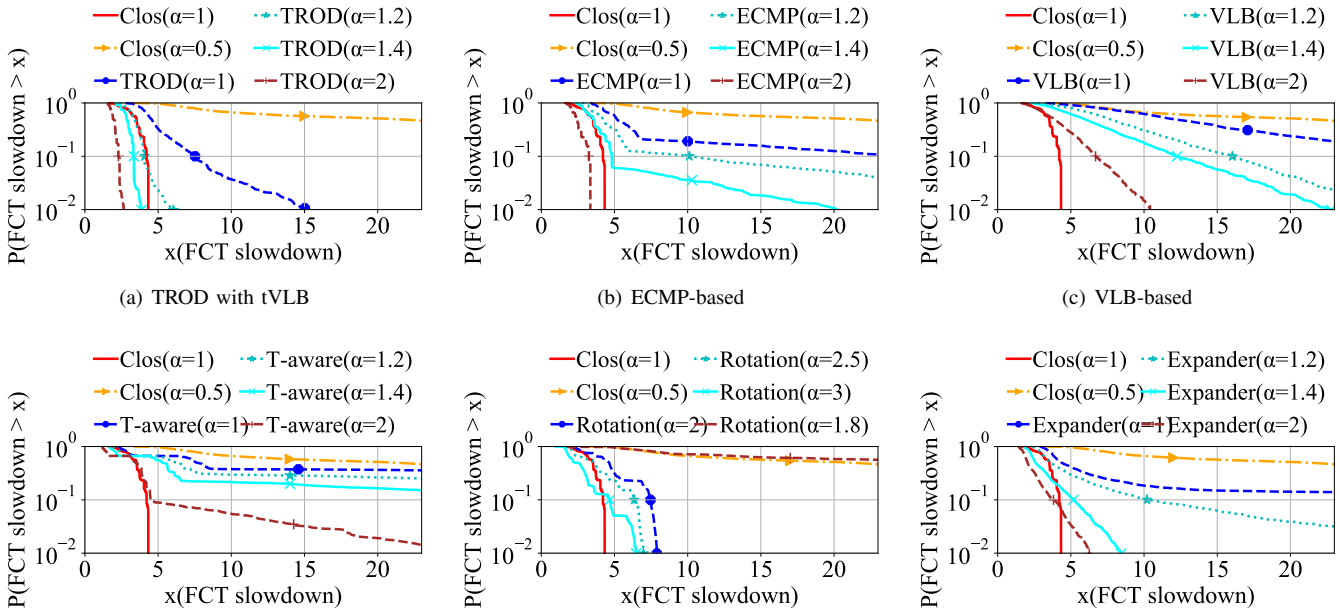
Clearly, TROD with tVLB routing performs the best. It starts outperforming the 1:1 Clos from $\alpha = 1.2$. When $\alpha = 1.4$, TROD is strictly better than the 1:1 Clos, reducing FCT by about $1.3\times$. The key to TROD's success is that, TROD can route the majority of packets through direct-hop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



(a) TROD with tVLB (b) ECMP-based (c) VLB-based

(d) Traffic-aware (a.k.a. Helios) (e) Rotation-based (f) Expander

Fig. 4. Performance under common traffic patterns. P(FCT slowdown>x) is the probability that the 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 matrix.



(a) TROD with tVLB (b) ECMP-based (c) VLB-based

(d) Traffic-aware (a.k.a. Helios) (e) Rotation-based (f) Expander

Fig. 5. Performance under synthetic bursty traffic patterns.

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, to achieve good FCT performance for the rotation-based approaches, the switch hardware, the congestion control and flow control schemes, etc., may 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.

*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.

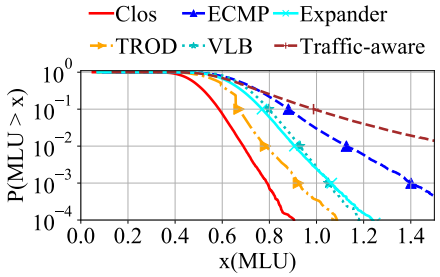
To synthesize busty traffic patterns, we take an arbitrary traffic pattern $D^b = [d^b_{ij}]$ from Facebook's trace as the base, and then add traffic burst on top of D^b. For any $i, j = 1, 2, ..., n$ and $i \neq j$, we create one bursty traffic pattern by increasing d^b_{ij} by certain amount of traffic 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 [20], [39].) 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. 5. 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 the 1:1 Clos.

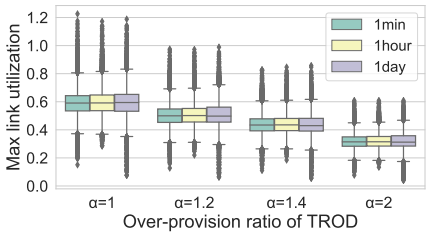
Other than TROD, the traffic-aware approach performs the worst, which cannot finish many flows even with $\alpha = 2$. The performance of the expander graph DCN also deteriorates. The expander graph DCN merely relies on routing to handle the skewed traffic patterns. As network load increases, this approach 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 $\alpha < 2$.

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. 6(a). Note that 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.

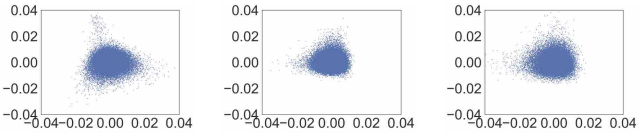


(a) Comparing MLU overflow probabilities.



(b) MLU comparison of TROD at different reconfiguration frequency.

Fig. 6. MLU performance evaluation.



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

Fig. 7. Visualizing traffic clusters of Facebook's trace using FastICA.

B. Dealing with Out-of-order Delivery

As described in §III-E, tVLB routing may cause out-of-order delivery of packets. We recommend enabling selective ack (SACK), to avoid retransmitting packets that arrived out-of-order. Indeed, DCTCP [40] and Swift [41] 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 [40], TCP BBR [42], Swift [41], etc. The reason is that packet reordering may not indicate a packet loss or network congestion under tVLB. If the TCP endpoints reduce the congestion window upon receiving three DACKs, the

TABLE II
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

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. II show the 90-percentile[5] 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).

C. TROD's Reconfiguration Frequency

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. 6(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 [43] for Facebook's one-day trace to visualize the trace's traffic clusters. Fig. 7 suggests that there is only one traffic cluster, and this traffic cluster does not change much with respect to time.

The observation in Fig. 7 applies to all the three Facebook's 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.

V. DISCUSSION

In the previous evaluation, we have demonstrated the performance benefits of TROD over other DCN architectures. As for the deployment complexity, TROD is clearly lower than the optical DCNs with frequent reconfigurations. However, when compared with static DCN architectures, e.g., Clos and expander, TROD still requires occasional OCS reconfiguration. It seems that this brings additional deployment overhead. However, we argue that having this capability of OCS reconfiguration may actually reduce the management complexity.

[5]A percent sign in the table indicates the percentile value of a set of data. e.g. **No DACK 90%** indicates the 90-percentile FCT slowdown value without reacting to DACKs.

As network demand grows gradually, data centers may require incremental expansion [26]. 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. However, the traditional Clos DCN may require significant labor work to manually reconnect the DCN topology, which is time-consuming and error-prone. One way to reduce the labor work of incremental expansion for Clos is to add an OCS layer between the PoDs and the core-layer spines. Then, an evolution path from Clos to TROD is formed.

After upgrading a Clos DCN with a layer of OCSs, TROD becomes a natural next step by removing the core-layer spines from the upgraded Clos. One may wonder how a data center communicates with the external world without the core-layer spines. This can be easily addressed by adding a special purpose PoD to peer with the external network, e.g., the border router in Google's DCN [7].

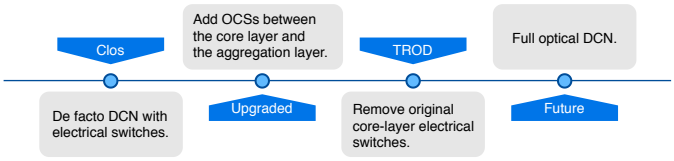


Fig. 8. Evolving from electrical DCN to optical DCN.

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, the ToR-layer DCN traffic exhibits even higher uncertainty. One possible solution is to construct a few groups of ToRs, and then study how to design a topology among ToR groups. We will study this problem in our future work.

VI. CONCLUSION

We proposed TROD, a traffic semi-agnostic PoD-level optical DCN, that achieves better FCT than the existing optical DCNs and the expander graph DCNs. With capacity overprovision 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. 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.

ACKNOWLEDGMENT

This work was supported by the NSF China under Grant 61902246 and the Program of Shanghai Academic/Technology Research Leader under Grant 18XD1401800.

REFERENCES

[1] Intel, "Affordably increase network bandwidth at 100 gbps and beyond," https://media20.connectedsocialmedia.com/intel/08/18818/Affordably_Increase_Network_Bandwidth_Optics.pdf, 2020.

[2] Sylex, "Where are the il limits for 10 40 and 100gbps applications," https://www.sylex.sk/sk/where-are-the-il-limits-for-10-40-and-100gbps-applications/.

[3] Juniper, "Migrating to a 40 gbps data center," https://www.juniper.net/us/en/local/pdf/whitepapers/2000578-en.pdf, 2015.

[4] H. Yu, P. Doussiere, D. Patel, W. Lin, K. Al-Hemyari, J. Park, C. Jan, R. Herrick, I. Hoshino, L. Busselle *et al.*, "400gbps fully integrated dr4 silicon photonics transmitter for data center applications," *OFC*, 2020.

[5] H. Ballani, P. Costa, R. Behrendt, D. Cletheroe, I. Haller, K. Jozwik, F. Karinou, S. Lange, K. Shi, B. Thomsen, and H. Williams, "Sirius: A flat datacenter network with nanosecond optical switching," in *SIGCOMM*, 2020.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM*, 2008.

[7] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano *et al.*, "Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network," in *SIGCOMM*, 2015.

[8] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, and S. Sengupta, "Vl2: A scalable and flexible data center network," in *SIGCOMM*, 2009.

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

[10] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, and M. Ryan, "c-through: Part-time optics in data centers," in *SIGCOMM*, 2010.

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

[12] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G. M. Voelker, G. Papen, A. C. Snoeren, and G. Porter, "Circuit switching under the radar with reactor," in *NSDI*, 2014.

[13] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer, "Firefly: A reconfigurable wireless data center fabric using free-space optics," in *SIGCOMM*, 2014.

[14] M. Ghobadi, R. Mahajan, A. Phanishayee, N. Devanur, J. Kulkarni, G. Ranade, P. A. Blanche, H. Rastegarfar, M. Glick, and D. Kilper, "Projector: Agile reconfigurable data center interconnect," in *SIGCOMM*, 2016.

[15] L. Chen, K. Chen, Z. Zhu, M. Yu, G. Porter, C. Qiao, and S. Zhong, "Enabling wide-spread communications on optical fabric with megaswitch," in *NSDI*, 2017.

[16] W. M. Mellette, R. Mcguinness, A. Roy, A. Forencich, and G. Porter, "Rotornet: A scalable, low-complexity, optical datacenter network," in *SIGCOMM*, 2017.

[17] W. M. Mellette, R. Das, Y. Guo, R. McGuinness, A. C. Snoeren, and G. Porter, "Expanding across time to deliver bandwidth efficiency and low latency," in *NSDI*, 2020.

[18] R. Zhang-Shen and N. McKeown, "Designing a fault-tolerant network using valiant load-balancing," in *INFOCOM*, 2008.

[19] C. Delimitrou, S. Sankar, A. Kansal, and C. Kozyrakis1, "Echo: Recreating network traffic maps for datacenters with tens of thousands of servers," in *IISWC*, 2012.

[20] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," in *SIGCOMM*, 2015.

[21] I. CALIENT Technologies, https://www.calient.net/.

[22] I. Agiltron, https://agiltron.com/.

[23] I. Polatis, https://www.polatis.com/.

[24] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosunblum, and A. Vahdat, "Exploiting a natural network effect for scalable, fine-grained clock synchronization," in *NSDI*, 2018.

[25] Y. Li, R. Miao, H. Liu, Y. Zhuang, F. Feng, L. Tang, Z. Cao, M. Zhang, F. Kelly, M. Alizadeh, and M. Yu, "Hpcc: High precision congestion control," in *SIGCOMM*, 2019.

[26] S. Zhao, R. Wang, J. Zhou, J. Ong, J. C. Mogul, and A. Vahdat, "Minimal rewiring: Efficient live expansion for clos data center networks: Extended version," in *NSDI*, 2019.

[27] T. Benson, A. Anand, A. Akella, and M. Zhang, "Microte: fine grained traffic engineering for data centers," *CoNEXT*, 2011.

[28] O. N. Fundation, "Openflow switch specification 1.3.1," ONF Technical Report, Tech. Rep., 2012.

[29] P. D. P. at Terabit Speeds, http://conferences.sigcomm.org/sigcomm/2018/files/slides/p4/P4Barefoot.pdf, 2018.

[30] E. L. Fernandes, E. Rojas, J. Alvarez-Horcajo, Z. L. Kis, D. Sanvito, N. Bonelli, C. Cascone, and C. E. Rothenberg, "The road to bofuss: The basic openflow userspace software switch," *Journal of Network and Computer Applications*, 2020.

[31] F. Chen, C. Wu, X. Hong, Z. Lu, Z. Wang, and C. Lin, "Engineering traffic uncertainty in the openflow data plane," in *INFOCOM*, 2016.

[32] L. J. Chaves, I. C. Garcia, and E. R. M. Madeira, "Ofswitch13: Enhancing ns-3 with openflow 1.3 support," in *Proceedings of the Workshop on ns-3*, 2016.

[33] K. Tantayakul, R. Dhaou, B. Paillassa, and W. Panichpattanakul, "Experimental analysis in sdn open source environment," in *ECTI-CON*. IEEE, 2017.

[34] 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*. IEEE, 2016.

[35] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," in *SIGCOMM Review*, 2006.

[36] Netbench, https://github.com/ndal-eth/netbench.

[37] A. Valadarsky, M. Dinitz, and M. Schapira, "Xpander: Unveiling the secrets of high-performance datacenters," in *HotNets*, 2015.

[38] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, "Xpander: Towards optimal-performance datacenters," in *CoNEXT*, 2016.

[39] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *SIGCOMM*, 2010.

[40] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data center tcp (dctcp)," in *SIGCOMM*, 2010.

[41] G. Kumar, N. Dukkipati, K. Jang, H. M. G. Wassel, X. Wu, B. Montazeri, Y. Wang, K. Springborn, C. Alfeld, M. Ryan, D. Wetherall, and A. Vahdat, "Swift: Delay is simple and effective for congestion control in the datacenter," in *SIGCOMM*, 2020.

[42] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Communications of the ACM*, vol. 60, pp. 58–66, January 2017.

[43] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, 2000.