Modeling Dynamic Patterns Adapted Joint Multi-dimension Resource Scheduling via Graph Sequence in Optical Data Center Network

Cen Wang
Photonic Transport Network Lab.
KDDI Research INC.
Saitama-ken, Japan
ce-wang@kddi-researh.jp

Takehiro Tsuritani
Photonic Transport Network Lab.
KDDI Research INC.
Saitama-ken, Japan
tsuri@kddi-research.jp

Xiong Gao
State Key Lab of IPOC
BUPT
Beijing, China
xionggao@bupt.edu.cn

Hongxiang Guo
State Key Lab of IPOC
BUPT
Beijing, China
hxguo@bupt.edu.

***Abstract*—Traffic patterns in data center network (DCN) may have distinguished features in a graph view. The traffic patterns may vary by time in DCN, and network topology reconstruction may help to adapt these dynamic traffic patterns. Traditional network scheduling may ignore the feature of patterns, which results in performance deteriorations. Optical DCN with multi-dimension resources may provide topology flexibility. If the multi-dimension resource scheduling problem is modeled as topology reconstruction by time, it may benefit the dynamic traffic patterns. To achieve this goal, this paper used a graph sequence to represent the time-varying network topology, and then a graph sequence based scheduling (GSS) algorithm has been proposed to optimize the traffic patterns. By simulation on our network modeling tool, the effectiveness on lowering latency and maximizing the link utilization of GSS has been verified.**

Keywords—traffic patterns adaption, multi-dimension resource scheduling, graph sequence

I. INTRODUCTION

In order to break through the capacity ceils of electrical packet switching (EPS), optical circuit switching (OCS) and optical packet switching (OPS) are actively investigated so as to be introduced into DCNs. Upon optical switching techniques, the optical DCN designs can be hybrid (e.g. OCS+EPS [1]) or all optical (e.g. OCS only [2][3] and OCS+OPS [4]) architectures, and the topology can be either tree-like or flattened. Optical switching brings extra dimensions of resources. For example, the OCS adds the wavelength dimension and the OPS gives the sub-wavelength (i.e. timeslots) dimension. In the near future, as the traffic in DCN continuously increasing, newly introduce multi-core and multi-mode switching may extend the dimensions. Moreover, it is easy to ignore that the optical switching can enable path reconfigurations by time in DCN. In whole network view, such path reconfigurations can lead to topology reconstructions, namely they can bring topology flexibility in DCNs.

Operators of data center networks (DCNs) are pursuing higher link utilization and lower latency (i.e. transmission time) to boost profits and provide better user experience. To achieve these two goals, traffic optimization via multi-dimension resource scheduling is an essential approach. Previously, traffic volumes have been paid much attention, especially the heavy-tailed distribution (i.e. the "80-20 rule") which the traffic volumes follow [5]. However, recent researches have revealed that not only the traffic volumes are various but also the traffic patterns are time-varying. Rather than volume distribution, the traffic pattern mentioned here indicates the features of whole network traffic in a graph view. The typical traffic patterns can be divided to globally random pattern (GR-pattern) and regionally clustered pattern (RC-pattern) [6]. In a GR-pattern, each network node pair may randomly have traffic or not, thus the GR-pattern shows the random graph. While the RC-pattern shows the cluster graph because that a group of network nodes with short distance (i.e. hops) have intra-group traffic but no inter-group traffic. Two patterns may alternatively appear by time.

A good topology can naturally optimize a traffic pattern due to less routings. Ideally, if there is traffic between a node pair and there happen to exist direct path between such the node pair, traffic will not travel long distance across the network, so the latency can be reduced due to lower communication costs and less congestions. Reconstructing topology may be a good choice to adapt dynamic traffic pattern. Optical DCN can naturally support the topology reconstruction. However, traditional idea neglects the traffic patterns and therefore the advantage of topology flexibility cannot be taken. The idea uses wavelength assignment (WA) [1][2] to offload huge traffic and dynamic bandwidth allocation (DBA) [7][8] to allot timeslots to forward small or medium traffic. Besides, such methods may decrease the link utilization because that the huge traffic may not arrive in most of the time.

To these ends, multi-dimension resource scheduling problem is better to be modeled as topology reconstructing for traffic patterns. Since the traffic patterns are time-varying, and all the traffic patterns along time axis are expected to be optimized, so a topology graph sequence scheduling (GSS) is proposed, as shown in Fig. 1. In our method, topologies are changed periodically to decrease communication costs and congestions, so that the latency can be lowered. Wavelengths

978-3-903176-21-8 (c) 2020 IFIP

and timeslots are both abstracted as the edges of topology graphs. The only difference is their differentiated lifetime. In this way, multi-dimension can be jointly scheduled, wavelengths are not only used for the huge traffic but are used on-demand, so that the high resource utilization can be guaranteed. We applied our GSS to multiple DCNs (i.e. FatTree, Clos and Lattice). By simulation, it has been verified that GSS can achieve as high as 3.2×, 3.37× and 3.65× acceleration of traffic transmission time on the three kinds of networks respectively.

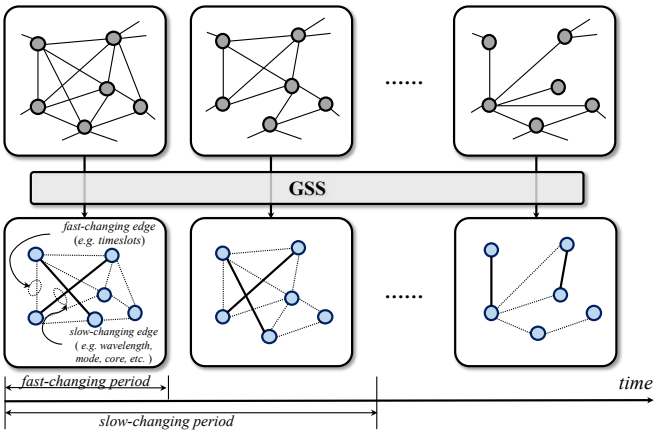

Fig. 1 The basic idea of the GSS

II. Related Work

A. Traffic measuring in DCN

The "80-20" rule is well-known distribution of traffic in DCN. It said that 20% traffic contributes 80% total volume. However, such distribution lacks the time information of the traffic. Normalized change [5] is used to investigate the dynamic traffic volume by time:

$$NC = \frac{|V_{t+1} - V_t|}{V_t}$$

where V_t is the volume of a traffic matrix (i.e. node pair traffic to form a matrix) at time t, and the NC is the normalized change. When the interval is 10s, the NC can vary from 0.37 to 1.49 [5]. Within a traffic matrix, traffic volume of node pairs is various, the volume can use heavy-tailed distribution to fit.

In addition, graph patterns of traffic are also be investigated. GR-pattern and RC-pattern are the typical patterns. Some pattern may be intermediate state of GR-pattern and RC-pattern. RC-pattern may come from parallel computing such as MapReduce in DCN, the dataset of MapReduce from Facebook [9] can verify this. Note that the pattern and the volume distribution are simultaneous, for example a traffic matrix may show a certain pattern, meanwhile the traffic volumes inside follow the heavy-tailed distribution.

B. Optical DCN Achitectures

In majority of optical DCN, optical switching is introduced in rack-level. Namely, in the network abstraction, a node is actually a top of rack (ToR) switch. As aforementioned, optical DCN can be tree-like or flattened. In the tree-like networks, Helios, C-through [10] and OSA are well-known. The OSA only uses OCS. But in the Helios and the C-through, some ports on a ToR are connected to an OXC, other ports are connected to an upper layer EPS switch, thus they are hybrid. In all the three architectures, connections between ToRs can be adjusted after reconfigurations of OXC, thus they are topology flexible.

The typical flattened DCNs are Mordia [7], OvS [3] and OpenScale [4]. The ring-like Mordia introduces very fast OPS and the OvS applies OCS to form a butterfly network. The Mordia can transiently change the connections of ToRs by allocating timeslots. In the OvS, a ToR can only connect another ToR in the same row or the same column. The OpenScale is kind of the lattice architecture implemented by OPS and OCS. ToRs are connected into multiple hexagon-based lattices. The OPS is used within each lattice, while the OCS can build path between inter-lattice ToRs. It can be seen that only the Mordia and the OpenScale can support topology flexibility. Although the GSS can be used on any topology flexible DCN, in this paper, we focus on the multi-dimension DCN architectures.

C. Scheduling in Optical DCN

Wavelength scheduling focuses on the adaption of traffic with large volume. The objective usually is to maximize throughput. Typical algorithm is *b-matching*, which is to find maximum matches from the biograph of traffic. The traffic adaptive topology reconstruction (TATR) [4] is another approach to pursue higher throughput with lower complexity. There is another round-robin like wavelength scheduling [11] in DCN with ultra-large scale. But the target of this scheme is not to promote any performance but to further release complexity of scheduling.

Sub-wavelength scheduling aims to adapt short-lived traffic. The traffic matrix scheduling (TMS) [7] in Mordia models the scheduling problem into a matrix decomposition problem. Another work uses sub-wavelength resources to adapt application flows via application-based DBA [8].

III. Graph Sequence Scheduling (GSS)

We use traffic matrix (TM) to represent the traffic request of the optical DCN. In this way, it is easy to learn the traffic pattern. Then we want to change the network topology to adapt a series of TMs in a period of the time. A traffic matrix may be sampled in a period T_{TM}, while the topology reconstruction period is T_{NW}. The network should take some time to forward the traffic, so the $T_{TM} > T_{NW}$. Here we set $\frac{T_{TM}}{T_{NW}} = M$.

Given a current matrix $\mathbf{TM}_C$, network topology is changed to avoid the traffic routings and the traffic congestions. Mathematically, to achieve this goal is to decompose the $\mathbf{TM}_C$ to a series of graphs. In each graph, the degree of a node equals to the number of the ports on a ToR switch. Besides, the weight of an edge will not exceed the capacity of a path within T_{NW}. Each edge is actually a fixed path so that the routing is avoided. A path will be used for traffic of a certain node pair (i.e. the preemptive forwarding), so that the congestion is avoided. It can be seen that each graph is actually a desired topology of the network. This series of graphs within T_{TM} are the topology reconstruction strategy for the traffic patterns.

The next problem is how to do the decompositions. Because in a traffic matrix, the traffic volumes of all node pairs follow the heavy-tailed distribution, that is to say small volumes are the

Identify applicable funding agency here. If none, delete this text box.

majority. If the huge traffic is firstly to be forwarded, then the small traffic will wait until the resources are released, the latency of small traffic will increase, thus the average latency will increase accordingly. So, our principle to reconstruct topology is to adapt small traffic preferentially. The other issue is how to manage multi-dimension resources. The multi-dimension resource reflects in the different types of ToR port. For example, in the Helios, there are two types of ToR ports, one type of ports uses timeslot resources because they are connected to an EPS switch, the other ports use wavelength resources because they are connected to OXC. The timeslot resources can be scheduled fast, but adjustment period of the wavelength layer resources is preferred to be long due to the slow switching of OXC device. In this case, once a light path is established, and the corresponding node pair still has traffic, the light path will not be changed. Besides, the timeslots resources are scheduled before the wavelength resources. As a consequence, the small traffic will be assigned to the timeslot resources while the wavelength resources will forward the huge traffic with high probability. Note that the wavelength resources will always be used if there is traffic, and the lifetime of a light path can be as long as possible. This can guarantee the high utilization and decrease the frequency of adjustment. Our GSS algorithm is shown as the following pseudocode.

Our algorithm uses a graph as the network topology reconstruction strategy to jointly schedule the multi-dimension resources. Further, a graph sequence helps to adapt dynamic traffic patterns and manage the lifetime of wavelength paths. We apply the GSS to multiple kinds of optical DCN by simulation, using the *Network Modeling Tool* (NMT) built by us.

Algorithm 1: Graph Sequence Scheduling (GSS)

Input: $\mathbf{TM}_1$ $\mathbf{TM}_2$ ... $\mathbf{TM}_S$
Interval of TM sampling T_{TM}
Network topology reconstruction period T_{NW}
$M = T_{TM}/T_{NW}$
Initial network graph $\mathbf{G}$
Output: **GS**

for $i, i = [1, 2, ..., S]$ **do**
  $\mathbf{TM}_c \leftarrow \mathbf{TM}_c + \mathbf{TM}_i$
  for $j, j = [1, 2, ..., M]$ **do**
    $\mathbf{G}_{i,j}$ = GraphSeqGen($\mathbf{G}, \mathbf{TM}_c$)
  end
end
return all $\mathbf{G}_{i,j}$ as **GS**

GraphSeqGen($\mathbf{G}, \mathbf{TM}_c$)
$\mathbf{G}_{i,j} \leftarrow \mathbf{G}$
N = NumberofNode(G)
$\mathbf{P} = [P_1, ..., P_k, .., P_N]$ = NumberofPort(G)
$\mathbf{TM}_c$ = UTM($\mathbf{TM}_c$)
▷ Transfer current matrix to upper triangular matrix
for $k, k = [1, 2, ..., N]$ **do**
  TrafficList = $\mathbf{TM}_c(k, k+1 : N)$
  while *True* **do**
    if $P_k = 0$ **then**
      break
    else
      VolumeSet = Sort(Set(TrafficList))
      for $l, l = 1, 2, ..., P_k$ **do**
        V = Length(VolumeSet)
        for $p, p = [1, ..., V]$ **do**
          $\mathbf{x}$, $\mathbf{x}_i$= FindTrafficVolume(TrafficList,VolumeSet[p])
          ▷ Find traffic volume equaling to VolumeSet[p] and its corresponding destination
          L = Length($\mathbf{x}$)
          $\mathbf{P}_\mathbf{x} = [P_{\mathbf{x}_i[1]}, ..., P_{\mathbf{x}_i[L]}]$
          if *not all elements in* $\mathbf{P}_\mathbf{x}$ *is 0* **then**
            dst = Max($\mathbf{P}_\mathbf{x}$)
            $P_{dst} -= 1$
            $\mathbf{G}_{i,j} \leftarrow$ Reconstruct($\mathbf{G}_{i,j}$, $[k, dst]$)
            break
          end
        end
      end
    end
  end
end
return $\mathbf{G}_{i,j}$

IV. Network Modeling Tool (NMT)

The NMT is written by python. It is light-weighted and easy to use. The NMT mainly includes three components: traffic generator, network builder and evaluation module. The source code of such tool can be found in https://github.com/wcdtom/NetworkModelingTool.

A. Traffic Generator

Traffic generator can generate traffic matrix with different traffic patterns. The traffic volumes inside a traffic matrix can also be determined. For example, in our work, the traffic volumes of a matrix are heavy-tailed distribution. Moreover, traffic generator can also produce a series traffic matrix to imitate time-varying traffic. The total traffic volumes of these matrixes are dynamic following the aforementioned normalized change.

B. Network Builder

The network builder can build multiple types of DCNs. Basically, the topology of network such as the FatTree, the clos and the lattice topologies can be generated. Further, switches, ports and queues can be established for scheduling study. This module can also transfer scheduling strategy to specific configurations of the switches, the ports and the queues, and such configurations can be sent to physical devices when connecting to an SDN controller.

C. Evaluation Module

Current evaluation module support latency, packet loss, link utilization and throughput. The latency is obtained based on queue length. Queue length in each node is calculated according to the traffic requests, network topology and the routing table. The link utilization is obtained according to the residual bandwidth of a link or the free time of a link.

Each of the component provides a lot of functions, users can use them to do the simulation or build an orchestrator. We also provides some real datasets of DC traffic or *AI+networking* examples on NMT.

V. Simulation and Results

Our simulation is based on the FatTree, the clos and the lattice optical DCN. The latency and link utilization are compared under GSS and traditional scheduling method.

A. Setups

We apply our GSS scheme to three types of DCNs. In the simulation, a ToR switch is regarded as a node, we leave the

sever level scheduling in the future. Each network scale is with 288 nodes (i.e. 288 ToRs). We set this number of nodes because it is easy to build each kind of network. In each network, a ToR has 12 uplink ports for electrical switching or sub-wavelength switching. Additionally, we adjust the number of uplink port to OXC from 1 to 6. The network oversubscription ratio is 1:1, all the link capacity is 10Gbps.

The first network type is the *k-ary* FatTree. In this architecture, there are $k = 24$ pods to connect severs. In a pod, there are $\frac{k}{2} = 12$ accessing switches and the same number of the aggregation switches. The accessing switch is the ToR switch. To connect all pods, there are $(k/2)^2 = 144$ core switches. In this case, the number of accessing switch (i.e. ToR switch) is $\frac{k^2}{2} = 288$.

The second type is the three-layer clos network. In the lowest layer, there are 288 ToRs. Then, the upper layer contains 72 switches with 48 downlink ports and 48 uplink ports (i.e. totally 96 ports). The top layer should have 48 switches, each switch needs 72 ports. We set this architecture because the commercial 96-port switch is not hard to find but switch with larger scale may not be available.

Lattice-based network is the last selected type, which is a typical representation among those flattened optical DCN architecture. A lattice can be a rectangular, a triangle or a hexagon. In order to guarantee the number of uplink ports is 12, a basic lattice is set as a hexagon. The 288-node network can be consisted of 122 hexagon cells. In fact, Zhang et al. [4] has reported such kind of hexagon-based scalable optical DCN and the corresponding node structure.

We set traffic normalized change is from 0.37 to 1.49, and the magnitude of initial traffic volume is 10^5 MB. 100 TMs are generated during 100s. Within a TM, the traffic volumes of node pairs follow the heavy-tailed distribution. (i.e. the pareto distribution). The patterns of those TMs are randomly changed with equivalent probability between global random style and regional cluster style. Simulations are operated 100 times, and we give the statistic results of latency and link utilization.

B. Latency

Our baseline is round-robin scheduling [11] of light paths in wavelength layer. After the light paths are established, traffic is forwarded via shortest path routing. The latency is given under such situation. We called TM latency to represent the completion time to forward all the traffic of a matrix. Because the absolute value of the TM latency is related to the magnitude of traffic volume, so the relative promotion ratio is preferred to be utilized. The relative promotion ratio uses the TM latency under baseline to divide the TM latency under modified scheduling [12]. In the simulation, the TM latency of the TATR [4] and the GSS are compared. The adjustment period of the round-robin scheduling and the TATR are set to T_{TM}.

In Fig. 2, it can be seen that, generally, the GSS gets higher promotion than the TATR. It may because GSS uses multi-dimension to adapt traffic patterns, But the TATR only build wavelength path for huge traffic, which caused the mismatch of topologies and traffic patterns. When the number of ports to OXC is increasing, the promotion ratio decreases. This is because, the added ports can provide more paths, which can reduce the network distance (i.e. hops between nodes). When network distance is lower, traffic can more likely be forwarded without both multiple hops routing and the congestions. Since the GSS aims to avoid the routings and the congestions, in such case, the optimization performance of the GSS deteriorates.

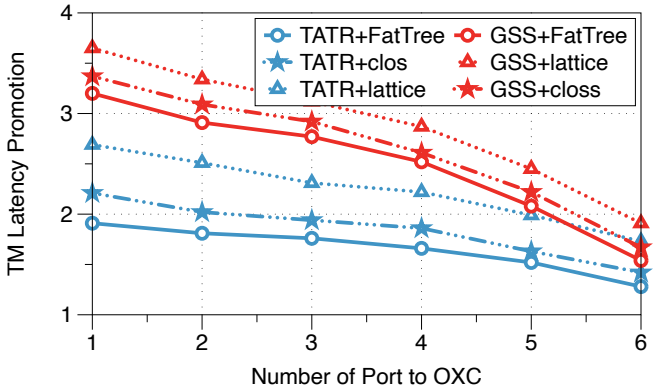

Fig. 2 The TM latency promotion of the TATR and the GSS

We also investigate how the GSS benefits the different traffic patterns. The results are shown in Fig. 3. On all the three types of networks, the TM latency of the RC-pattern has been promoted greatly under the GSS. But the promotion of the GR-pattern under the GSS is lower. When using the TATR, the promotions of two type of patterns are not so different. It can be concluded that our GSS is better for the RC-pattern.

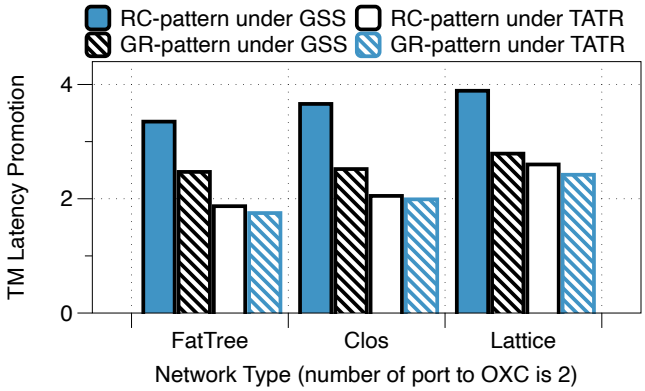

Fig.3 The TM latency promotion of the GR-pattern and the RC-pattern under the TATR and the GSS

C. Link Utilization

The link utilization on the three kinds of networks are depicted in Fig. 4. We also use the relative value, because the utilization is related to the traffic volume and the link capacity. The baseline is also the round-robin scheduling. The promotion ratio uses average link utilization of GSS or TATR to divide that of the baseline. The GSS gets higher utilization promotion than the TATR, but the promotions are not so difference among the networks under a certain scheduling. The link utilization of the TATR is not so high which may because the adjustment period is long, and the wavelength path may be idle for some time. If the adjustment period is set shorter, the link utilization may be promoted. Even so, the TATR does not jointly schedule the

multi-dimension resources, the utilization of timeslot resources cannot be guaranteed.

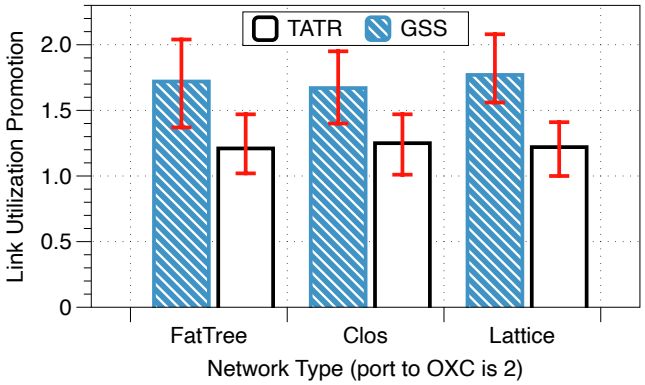

Fig.4 The link utilization promotion under the TATR and the GSS

CONCLUSION

We have proposed the GSS algorithm to jointly schedule the multi-dimension resources to adapt time-varying traffic patterns. Due to the better coordination of multi-dimension resources, the GSS can use topology reconstruction modeling to match various traffic patterns. Meanwhile, the latency of traffic can be reduced, and the link utilization can be promoted. On the NMT, we verified the GSS on three types of optical DCNs. The results show that the GSS can outperform the TATR in the traffic latency and the link utilization promotions. In addition, the GSS can benefit the RC-pattern more.

REFERENCES

[1] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen and A. Vahdat, "Helios: a hybrid electrical/optical switch ar-chitecture for modular data centers," in *Proc. of ACM SIGCOMM 2010 Conference*, pp. 339-350, August 2010.

[2] K. Chen, A. Singla, K. Ramachandran, L. Xu, Y. Zhang, X. Wen and Y. Chen, "OSA: An Optical Switching Architecture for Data Center Networks With Unprecedented Flexibility," IEEE/ACM Trans. Networking, vol. 22, no. 2, pp. 498-511, Apr. 2014.

[3] Z. Zhu, S. Zhong, L. Chen and K. Chen, "Fully programmable and scalable optical switching fabric for petabyte data center," *Optics Express*, vol. 23, no. 3, pp. 3563-3580, 2015.

[4] Dongxu Zhang, Jian Wu, Hongxiang Guo, Rongqing Hui. "Optical switching based small-world data center network." Computer Communications, 103.C, pp.153-164, 2017.

[5] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *Proc. of the ACM SIGCOMM 2009 Conference*, pp. 202-208, November 2009.

[6] Yiting Xia, Xiaoye Steven Sun, Simbarashe Dzinamarira, Dingming Wu, Xin Sunny Huang, T. S. Eugene Ng. "A Tale of Two Topologies: Exploring Convertible Data Center Network Architectures with Flat-tree," in *Proc. of ACM SIGCOMM 2017 Conference*, pp. 295-308, August 2017.

[7] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, and A. Vahdat, "Integrating microsecond circuit switching into the data center," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 447-458, 2013.

[8] Cen Wang, Hong Cao, Shenzhen Yang, Junyuan Guo, Hongxiang Guo and Jian Wu, "Decision Tree Classification based Mix-flows scheduling in Optical Switched DCNs," *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1-3, 2018.

[9] [Online]. Available: https://github.com/coflow/coflow-benchmark.

[10] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan, "c-Through: Part-time optics in data centers," in *Proc. of the ACM SIGCOMM 2010 Conference*, pp. 327-338, August 2010.

[11] W. M. Mellette, R. McGuinness, A. Roy, A. Forencich, G. Papen, A. C. Snoeren, and G. Porter, "Rotornet: A scalable, low-complexity, optical datacenter network. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication," in *Proc. of ACM SIGCOMM 2017 Conference*, pp. 267-280, August 2017.

[12] M. Chowdhury, Y. Zhong, and I. Stoica, "Efficient coflow scheduling with varys," *in Proc. of the ACM SIGCOMM 2014 Conference*, pp. 443-454, August 2014.