Joint Static and Dynamic Traffic Scheduling in Data Center Networks

Zizhong Cao
New York University
New York, NY 10012, USA
Email: zc347@nyu.edu

Murali Kodialam and T.V. Lakshman
Bell Laboratories, Alcatel-Lucent
Crawford Hill, NJ 07733, USA
Email: muralik@alcatel-lucent.com, t.v.lakshman@alcatel-lucent.com

***Abstract*—The advent and continued growth of large data centers has led to much interest in switch architectures that can economically meet the high capacities needed for interconnecting the thousands of servers in these data centers. Various multi-layer architectures employing thousands of switches have been proposed in the literature. We make use of the observation that the traffic in a data center is a mixture of relatively static and rapidly fluctuating components, and develop a combined scheduler for both these components using a generalization of the load-balanced scheduler. The presence of the known static component introduces asymmetries in the ingress-egress capacities, which preclude the use of a load-balanced scheduler as is. We generalize the load-balanced scheduler and also incorporate an opportunistic scheduler which sends traffic on a direct path when feasible to enhance the overall switch throughput. Our evaluations show that this scheduler works very well despite avoiding the use of a central scheduler for making packet-by-packet scheduling decisions.**

I. Introduction

One of the major current challenges in networking is the efficient interconnection of the thousands to tens-of-thousands of servers deployed in a data center. This interconnection is typically a multi-layer Fat-Tree or Clos network. The processors are deployed in racks. Each rack can have a top-of-rack (ToR) Ethernet switch that is used to connect to an aggregation layer which consists of large Ethernet switches. And then the aggregation layer is connected to a set of routers which provide connection to the public Internet. Most current designs deploy electrical switches and routers. The massive scale, high capacities and large bisection bandwidth needed makes the design of the switching architecture for this interconnect challenging. Several other factors contribute to making the design a challenge as well:

1) The distributed nature of the interconnect (spanning a data center and not limited to a rack).
2) The need for low costs given the very large number of ports to be interconnected.
3) Limited predictability in traffic patterns.

This last factor is exacerbated in multi-tenant data centers where users can dynamically use a part of the data center's resources (as with Amazon's EC2 and other cloud-computing services). With the set of users dynamically changing over time, the traffic patterns can correspondingly change with little a priori predictability. The exception is traffic from certain applications (such as routing parallel computing tasks) or long-lived users whose traffic can have static non-varying (over short time-scales) components. This unpredictability is often handled by over-provisioning network capacities. However, this raises networks costs and energy consumption. A desirable goal is to devise schedulers that can handle traffic uncertainty without significant over-provisioning of network capacities in the data center.

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

There have been several recent proposals [1], [2] to deploy optical switches to interconnect the ToRs in order to reduce the cost and complexity of running the data center networks. (See [3] for a survey of different optical interconnect technologies for data centers). Several optical architectures have been proposed including (2-D and 3-D) MEMS [4] based optical switches and broadcast-select optical rings. In the case of MEMS, a reconfigurable mirror is used to switch the incoming beam. In a broadcast-select architecture, each input node transmits with a fixed wavelength and the receiver tunes to that wavelength to get data from the transmitter. The reconfiguration time for MEMS can be in the order of microseconds and for the broadcast-select ring can be in the order of tens of nanoseconds. Therefore these optical architectures can be used to switch traffic in relatively fast time scales.

It has been observed in recent studies [5], [6], [7] that a part of the traffic between ToRs and hence between aggregation routers comprises of a relatively static component and a highly dynamic component. The static component is due to the assignment of VMs in closely interconnected racks and therefore there is steady traffic between these racks independent of the applications running in the data center. The highly variable dynamic component is due the presence of middleboxes in the network that provide functionality for the applications. Traffic has to be routed to and from middleboxes and this traffic is highly application-dependent and can vary rapidly across time. The relatively static component of the traffic varies over seconds, while the dynamic traffic varies over milliseconds or smaller time scales. This has led to the development of a hybrid approach [1], [2], [4], [5]. The static portion of the traffic is scheduled using pre-provisioned optical circuits, while the variable portion of the traffic is carried by electrical packet switches. However, this kind of design suffers from the extra costs of deploying two different switching systems. Also, the complexity in setting up the optimal circuits according to the changing traffic pattern cannot be overlooked.

Besides the hybrid design that combines the strengths of electrical packet switching and optical circuit switching, efforts have also been made to all-optical systems. OSMOSIS [8] adopts semiconductor optical amplifiers (SOA) and a broadcast-and-select architecture for fast switching, but suffers from high cost and energy consumption. Data Voltex [9] resolves the need of fast switching and reconfiguration by composing a banyan network, and allows flexible deflection routing to resolve contentions. However, the wiring complexity becomes intimidating when the system scales, and the packet delay could be indeterminent due to uncertain number of deflections. DOS [10] employs Arrayed Waveguide Grating Router (AWGR) for fast switching and a loopback shared buffer for contention resolution, but relies on high speed-ups at the outputs and shared memory for non-blocking switching.

Viewing the limitations of these systems, we propose a new design that significantly differs from the existing ones. As we shall see in latter parts of this paper, our design deploys a single interconnection network for both static and dynamic traffic, thus avoiding the extra costs of implementing and maintaining two different switching systems. Also, such a unified system easily adapts to changing portions of static and dynamic traffic. The software complexity and hardware requirement are also lowered by adopting simple wiring and distributed scheduling for the interconnection network.

A. Scheduling Challenges

There are many aspects of data centers that pose unique challenges to developing scheduling algorithms that are different from standard router scheduling. A router or switch designed to operate in a chassis or rack can employ a centralized scheduler to compute a match between ingress and egress nodes in each transmission time slot [11]. This approach is quite difficult to implement efficiently in a data center due to the following:

- The aggregation switch has to collect traffic from the top-of-rack switches which are distributed in the data center. Unlike a traditional router where all line cards are located on the same chassis (except for high-end multi-chassis routers), the line cards for switches in the aggregation layer are not all constrained to be on one chassis. This significantly complicates the implementation of a packet by packet centralized scheduler due to the delays in both collecting and disseminating information.
- The increasing line speeds as well as the number of top-of-rack switches that have to be interconnected makes centralized computation increasingly infeasible.

Instead of scheduling the static and dynamic traffic separately, we instead develop an approach that jointly schedules both traffic classes optimally without a packet-by-packet centralized scheduler. The scheduler instead runs at the time scales over which static traffic varies.

B. Our Contributions

There are two extreme solutions to scheduling an interconnection network. At one end of the spectrum is the centralized Birkhoff-von Neumann scheduler that can be used to schedule static traffic [11]. The other extreme is the load balanced scheduler [12] that uses over-provisioning to eliminate the need of centralized scheduling. The main contribution of this paper is the development of a Generalized Load Balanced Scheduler (GLOBE) that bridges the Birkhoff-von Neumann scheduler and the load balanced scheduler. GLOBE comprises of two modules:

- The **Static Traffic Scheduling** module that schedules the static traffic matrix while maximizing the throughput of the worst case dynamic traffic.
- The **Distributed Cycle Canceling** module that is implemented at the input nodes to improve the throughput and delay performance of the schedule.

As part of GLOBE we have developed:

- A simple algorithm for optimally allocating the traffic split ratios in generalized load balanced switch to maximize the worst case switch throughput.
- A decentralized local information cycle canceling mechanism that can be used to further increase throughput.
- A primal-dual approximation algorithm for optimal cycle canceling that uses a sequence of minimum mean cycles as the best cycles to cancel.
- A simple and novel counter based algorithm for practically implementing the cycle canceling scheme.

The cycle canceling algorithm is of independent interest and can be used to improve the performance of any load balanced switching architecture. We want to point out that packets can arrive out of sequence at the egress of a load balanced switch. This problem has been studied extensively and there are several solutions [13], [14], [15], [16] to handle the packet reordering problem in load balanced switches. Any one of these approaches can be used in combination with the scheduling algorithms developed in the paper in order to avoid out-of-sequence packets.

II. System Model

Consider an interconnection network that connects a set of ToR switches or a set of aggregation routers in a data center. Typically. there can be hundreds of ToR switches and perhaps tens of aggregation routers in current data centers. The techniques developed in this paper apply to both electronic and optical interconnects. However, in order to fix a model, we assume that the interconnect network is a WDM broadcast-select optical ring [3], [17] interconnecting a set of aggregate routers. Each element being interconnected is viewed as a node in the ring. Each node has a fixed wavelength transmitter

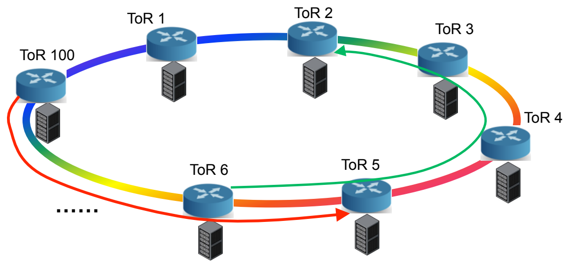

Fig. 1. Broadcast-Select WDM Ring

and a tunable receiver. The receiver tunes to the appropriate wavelength to receive data from a transmitting node. The receiver can switch from one wavelength to another in the order of tens of nanoseconds. This gives the switch fast switching times. Using current WDM technology (carrying 100 wavelengths) we can interconnect 100 nodes in a ring. Each wavelength can carry over 100 Gb/s making the broadcast select ring a 10 Tb/s crossconnect. We want to stress the fact that the techniques developed in this paper have wider applicability than the broadcast-select architecture. It can be used in MEMS based switching architectures or even to high speed electronic interconnects as well.

A. *Traffic Model*

The traffic matrix between the ToR or the aggregation routers has two components:

- The portion of traffic that varies slowly will be referred to as **static traffic**, and represented by matrix $\mathbf{S}$, where s_{ij} is the known traffic between nodes i and j. The static traffic varies relatively slowly over time (order of seconds). We assume that the static traffic matrix is known (or can be estimated accurately) at any given point of time.
- The unknown and fluctuating traffic will be referred to as **dynamic traffic**, and represented by matrix $\mathbf{D}$, where d_{ij} is the dynamic traffic between nodes i and j.

Let n denote the number of elements that are interconnected. Traffic in the interconnect network is given in the form of a $n \times n$ traffic matrix $\mathbf{T}$, where t_{ij} is the instantaneous traffic between nodes i and j. Assume that the ring speed is normalized to one. Given a traffic matrix $\mathbf{T}$, we define the the *rate* of a traffic matrix as the maximum of the row and column sums of the traffic matrix. The rate of the traffic matrix $\mathbf{T}$ is denoted by $\mathcal{R}(\mathbf{T}) = \max_i \left\{ \sum_j t_{ij}, \sum_j t_{ji} \right\}$. It is well known [11] that a traffic matrix $\mathbf{T}$ is schedulable if and only if $\mathcal{R}(\mathbf{T}) \leq 1$.

If $\mathcal{R}(\mathbf{T}) \leq 1$ and if $\mathbf{T}$ is relatively static, then it is easy to use a Birkhoff-von Neumann (BV) decomposition of the traffic to generate a schedule. In the BV decomposition, the traffic matrix is decomposed into permutation (switching) matrices, i.e., $\mathbf{T} = \sum_i \phi_i \mathbf{P}_i$ where ϕ_i is positive with $\sum_i \phi_i = 1$ and $\mathbf{P}_i$ is a permutation matrix. The BV decomposition involves solving a sequence of bipartite matching problems and each of these can be solved in $O(n \log n)$ time (See [18]).

Therefore, the BV decomposition can be used to schedule the static traffic matrix $\mathbf{S}$ if $\mathcal{R}(\mathbf{S}) \leq 1$. However, the BV decomposition based approach is not suitable for dynamic traffic since the traffic matrix $\mathbf{D}$ is not known ahead of time. Neither are standard switch scheduling techniques like finding the maximum weight matching. This is due to the fact that the traffic at the nodes is distributed through the data center, and it is not easy to communicate the virtual output queue (VOQ) lengths to a central place in order to compute the (exact or approximate) max weight matching as it is done in standard routers. Moreover, the increasing speed of the interconnect makes maximum matching computation at the end of each time slot increasingly infeasible.

B. *Scheduler Structure*

The scheduler that we develop for the joint scheduling of static and dynamic traffic is called the Generalized Load Balanced Scheduler (GLOBE). The GLOBE scheduler comprises of two modules.

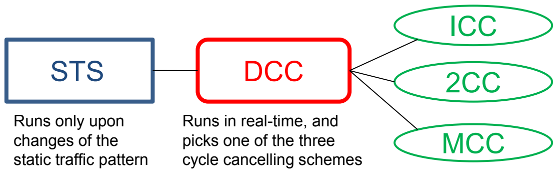
STS
DCC
ICC
2CC
MCC
Runs only upon changes of the static traffic pattern
Runs in real-time, and picks one of the three cycle cancelling schemes

Fig. 2. Scheduler structure.

- The **Static Traffic Scheduling (STS)** module is used to schedule the static portion of the traffic while maximizing the throughput of the worst case dynamic traffic. The main idea behind the static traffic module is a joint scheduling and BV decomposition for both static and dynamic traffic.
- The second module is the **Distributed Cycle Canceling (DCC)** module that runs at each ingress node independently in order to improve the throughput and delay performance.

In the rest of the paper, we describe the two modules in more detail.

III. GLOBE-STS: The Static Traffic Scheduling Module

GLOBE-STS is the module that schedules the static portion of the traffic while maximizing the throughput of the worst case dynamic traffic. The static portion of the traffic is specified by the traffic matrix $\mathbf{S}$ and is known ahead of time. The static portion of the traffic will be directly routed from the ingress node to the egress node by GLOBE-STS. For the dynamic portion of the traffic, GLOBE-STS uses a generalized asymmetric load balanced scheduler. The symmetric load balanced switch was first studied in [12] with the idea of eliminating the centralized scheduler. The symmetric load balanced scheduler will not work well in the presence of static traffic. We therefore generalize the load balanced scheduler for the case where there are both static and dynamic traffic. In the generalized load balanced scheduler, each node j has associated with it a fraction α_j with $\sum_j \alpha_j = 1$, and *every* ingress node sends a fraction α_j of its traffic through intermediate node j independent of the egress node of the packet. This can be thought of as a two-phase switch:

- **Phase 1:** Each ingress node sends a fraction α_j of its traffic to intermediate node j independent of the final destination.
- **Phase 2:** The intermediate node then forwards the packet to the destination node.

The α_j variables are called the *traffic split parameters*, and these are picked carefully to maximize the amount of dynamic traffic that can be carried by the switch.

The symmetric load balanced scheduler has $\alpha_j = \frac{1}{n}$ for all j. In the standard load balanced scheduler it can be shown (see [12]) that both the first and second phases can use a simple round robin switching pattern that is fixed ahead of time *independent of the traffic pattern* and therefore there is no need for a centralized scheduler. A traffic matrix $\mathbf{T}$ is schedulable using a load balanced scheduler if and only if $\mathcal{R}(\mathbf{T}) \leq \frac{1}{2}$.

For GLOBE-STS we want to pick splitting factors α_j that is good for the worst case dynamic traffic matrix. For picking the optimal set of α_j, we restrict our attention to dynamic traffic matrices $\mathbf{D}$ such that $\mathcal{R}(\mathbf{S}+\mathbf{D}) \leq 1$. For a fixed $\mathbf{S}$, the set of dynamic traffic matrices $\mathbf{D}$ are defined to belong to the set $\mathcal{D}_{\mathbf{S}}$ if and only if $\mathcal{R}(\mathbf{S}+\mathbf{D}) \leq 1$. For a fixed $\mathbf{S}$, let $\mathcal{D}_{\mathbf{S}}$ represent the set of dynamic traffic matrices that can be feasibly routed by a centralized algorithm. Without loss of generality, we assume that $s_{ii} = d_{ii} = 0$ for all i, so that there is no loopback traffic. Let $R_i = 1 - \sum_j s_{ij}$ and $C_i = 1 - \sum_j s_{ji}$ denote the capacity left over for the dynamic traffic, then $\mathbf{D} \in \mathcal{D}_{\mathbf{S}}$ if

$$\sum_j d_{ij} \leq 1 - \sum_j s_{ij} = R_i, \quad \forall i \tag{1}$$

$$\sum_j d_{ji} \leq 1 - \sum_j s_{ji} = C_i, \quad \forall i \tag{2}$$

In GLOBE-STS, we first characterize the total traffic between two nodes in the interconnect for *any* $\mathbf{D} \in \mathcal{D}_{\mathbf{S}}$:

- The amount of Phase 1 traffic between i and j is $\sum_k \alpha_j d_{ik} = \alpha_j \sum_k d_{ik} \leq \alpha_j R_i$.
- The amount of Phase 2 traffic between nodes i and j is $\sum_k \alpha_i d_{kj} = \alpha_i \sum_k d_{kj} \leq \alpha_i C_j$.

Therefore the total amount of traffic routed between nodes i and j is bounded by $\alpha_j R_i + \alpha_i C_j$. Note that the upper limit of the traffic between nodes i and j is independent of the actual dynamic traffic matrix $\mathbf{D}$. We want to pick α_j such that its worst case performance is optimized.

Theorem III.1. *Consider a generalized load balanced switch with fixed traffic matrix* $\mathbf{S}$ *and dynamic traffic matrix* $\mathbf{D}$*. The worst case scaling factor for the generalized load balancing algorithm is*

$$\theta_{\mathbf{D}}^{GLOBE} = \frac{\sum_{i \in P} \frac{\min\{R_i, C_i\}}{T - R_i - C_i}}{1 + \sum_i \frac{\min\{R_i, C_i\}}{T - R_i - C_i}} \tag{3}$$

where $T = \sum_i R_i$ *for* any *dynamic matrix* $\mathbf{D}$ *such that* $\mathcal{R}(\mathbf{S}+\mathbf{D}) \leq 1$.

Proof. If the dynamic traffic matrix is scaled by θ, then the amount of dynamic traffic between nodes i and j is $\theta\alpha_j R_i + \theta\alpha_i C_j$. In order for the dynamic scheduler to be stable, none of the ingress or egress nodes can be over-subscribed. Therefore

$$\sum_{j \neq i} [s_{ij} + \theta\alpha_j R_i + \theta\alpha_i C_j] \leq 1, \quad \forall i \tag{4}$$

$$\sum_{j \neq i} [s_{ji} + \theta\alpha_i R_j + \theta\alpha_j C_i] \leq 1, \quad \forall i \tag{5}$$

Noting that $\sum_j s_{ij} = 1 - R_i$, $\sum_j s_{ji} = 1 - C_i$, and representing the total amount of left-over capacity by $\sum_i R_i = \sum_i C_i = T$. From Inequalities (4) and (5), we can write

$$\alpha_i \leq \frac{1-\theta}{\theta} \frac{\min\{R_i, C_i\}}{T - R_i - C_i}. \tag{6}$$

Summing up the inequalities for all the α_i, and noting that $\sum_i \alpha_i = 1$, we solve for the worst case scaling factor θ^* as

$$\theta^* = \frac{\sum_{i \in P} \frac{\min\{R_i, C_i\}}{T - R_i - C_i}}{1 + \sum_{i \in P} \frac{\min\{R_i, C_i\}}{T - R_i - C_i}} \text{ and } \alpha_i^* = \frac{1-\theta^*}{\theta^*} \frac{\min\{R_i, C_i\}}{T - R_i - C_i}. \tag{7}$$

□

Once the optimal split ratios have been determined as per Equation (7), GLOBE-STS uses a BV decomposition to determine the schedules. An outline of GLOBE-STS for a given static matrix $\mathbf{S}$ is shown in Figure 7. Note that the computation of the optimal traffic split ratios is done only when the static traffic matrix $\mathbf{S}$ changes. There is no computation in each slot. Once the decomposition is known, each node (using a consistent algorithm) determines the next permutation to use and all the nodes implement this permutation (matching). We now describe the second module of GLOBE which is the Distributed Cycle Canceling module GLOBE-DCC.

IV. GLOBE-DCC: The Distributed Cycle Canceling Module

In GLOBE-STS we assumed that the ingress node sends packets to intermediate ports in the fixed ratio independent of the egress port. Since the ingress node knows the egress node for all the packets that originates from it, it can take advantage of this fact to improve the performance of the interconnect network. Instead of routing obliviously to the final destination, it is possible to improve throughput if the ingress node opportunistically sends packets in one hop to the egress node, thus reducing the traffic load in the second phase. This

GLOBE-STS: STATIC TRAFFIC MODULE

Input: Static matrix $\mathbf{S}$
Output: Optimal split factors α_j.

1) Compute R_i and C_i for each node. (Equations (1,2))
2) Determine the optimal split ratio. (Equation (7)).
3) Compute the flow between nodes i and j and compute traffic matrix $\mathbf{T}$ where
$$t_{ij} = s_{ij} + \alpha_j R_i + \alpha_i C_j.$$
4) Run the BV decomposition algorithm to determine the permutation matrices $\mathbf{P}_i$ and weights ϕ_i.
5) In each time slot pick the permutation matrix $\mathbf{P}_i$ with probability ϕ_i and use this matrix to set up the interconnects.

Fig. 3. Description of GLOBE-STS

step called cycle canceling has to be done carefully respecting the capacities that are allocated by GLOBE-STS. This idea of cut-through routing in a simple load balanced network was explored in [19]. However their approach does not extend to generalized load balancing. Further, they do not consider the dynamic implementation of cut-through that we outline in Section IV-C.

A. GLOBE-2CC: Two Cycle Canceling

We first illustrate cycle canceling using a simple two arc example. Consider input node i. At the end of GLOBE-STS, the amount of Phase 2 traffic generated between ports j and k by traffic originating at node i is the following:

- Of the traffic d_{ik} that has to be sent from node i to node k, $\alpha_j d_{ik}$ is sent to node j in Phase 1, and this in turn is sent to port k from node j in Phase 2.
- Of the traffic d_{ij} from i to j, a traffic of $\alpha_k d_{ij}$ is sent to node k in Phase 1, and thus is transferred from node k to node j in Phase 2.

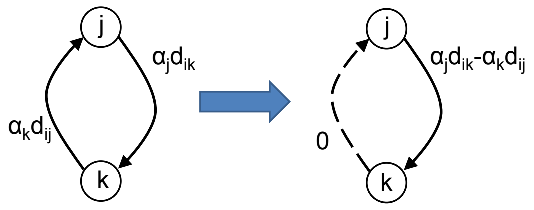

Fig. 4. Two Arc Cycle Canceling

Assume that node i can estimate d_{ij} and d_{ik}. Further assume without generality that $\alpha_j d_{ik} \geq \alpha_k d_{ij}$. Instead of sending $\alpha_k d_{ij}$ units of traffic through $i \to k \to j$, suppose node i delivers this traffic directly over $i \to j$ while transferring the same amount of traffic directly over $i \to k$ instead of sending it through j. We know that the second transfer can be made since $\alpha_j d_{ik} \geq \alpha_k d_{ij}$. This process of a two-way swap is shown in Figure 4. This cycle canceling will reduce the load in Phase 2 between ports j and k.

Apart from the ease of implementing two-way swaps, it is also easy to characterize the capacity region of GLOBE-2CC in closed form. The proof of this result follows directly from the argument that we have made above and is omitted in the interest of space.

Theorem IV.1. *Consider GLOBE with a static traffic matrix* $\mathbf{S}$. *Corresponding to this matrix is a split ratio* α_j. *GLOBE-2CC will be stable with a dynamic traffic matrix* $\mathbf{D}$ *if*

$$\sum_{j\neq i}\left[s_{ij} + \sum_k a_j d_{ik} + \sum_k [\alpha_j d_{ki} - \alpha_i d_{kj}]^+\right] \leq 1, \quad \forall i \quad (8)$$

$$\sum_{j\neq i}\left[s_{ji} + \sum_k \alpha_i d_{kj} + \sum_k [\alpha_i d_{kj} - \alpha_j d_{ki}]^+\right] \leq 1, \quad \forall i \quad (9)$$

where $[x]^+ = \max\{0, x\}$.

Note that in the case of symmetric traffic, with GLOBE-2CC we get 100% throughput as opposed to 50% using GLOBE-STS. This clearly is an extreme example, however there is *significant* performance improvement over GLOBE-STS even for the non-symmetric case. Though cycle canceling does not need any central coordination, each ingress node has to estimate the rates to each egress node and implement the cycle canceling algorithm. In Section IV-C, we show that this can be done without any explicit rate estimation. The idea is to use a counter at each ingress node i to automatically track $[\alpha_j d_{ik} - \alpha_k d_{ij}]^+$ and hence it is easy to determine where traffic has to be routed.

B. GLOBE-MCC: Maximum Cycle Canceling

The two arc cycle canceling heuristic can be easily extended to multi-arc cases, as shown in Figure 5.

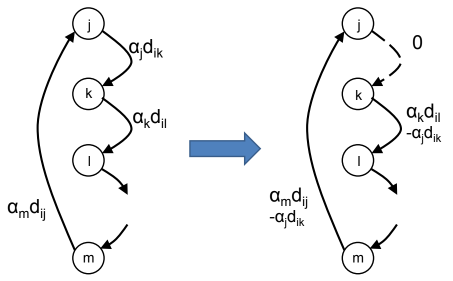
j
k
l
m
$\alpha_j d_{ik}$
$\alpha_k d_{il}$
$\alpha_m d_{ij}$
j
k
l
m
0
$\alpha_k d_{il}$ $-\alpha_j d_{ik}$
$\alpha_m d_{ij}$ $-\alpha_j d_{ik}$

Fig. 5. Multi-arc Cycle Canceling

We now formulate the problem of determining the set of cycles to cancel in order to get the maximum reduction in Phase 2 traffic. This optimization is done independently by each node using only the traffic that it can estimate locally. Consider a node i. It minimizes Phase 2 traffic that it generates by first constructing a Phase 2 flow graph, and then solving an optimal fractional cycle cover problem on this flow graph.

Node i estimates the d_{ij} values in order to construct the Phase-2 flow graph. This estimation is easy to do since all the VOQs for $i \to j$ traffic sits at node i. The flow graph at node i consists of $n-1$ nodes each representing an egress node. There is an arc between each pair of nodes in the graph. The capacity of the (directed) arc between ports j and k is $\alpha_j d_{ik}$ and represents the Phase 2 traffic scheduled by GLOBE-STS between j and k that originated at node i in Phase 1.

Let $\mathcal{C}$ represent the set of all cycles in the graph and c represent a generic cycle. Whenever we assign some flow Δ along a cycle c with $|c|$ arcs, the Phase 2 traffic can reduce correspondingly by $|c|\Delta$ according to the heuristic in Fig. 5. Denote by x_c the total amount of flow allocated over cycle c. Then the general problem is to determine the optimal flow allocations on these cycles that maximizes the amount of Phase 2 traffic reduction, subject to the arc capacity constraints. In order to keep notations simple, we use u_e to denote the capacity of arc e in the flow graph. Note that $u_e = \alpha_{a(e)} d_{ib(e)}$ where $a(e)$ is the tail of arc e and $b(e)$ is the head. The objective function reflects the fact that longer cycles eliminate more Phase 2 traffic. The problem can be formulated as follows:

$$\max F(\mathbb{X}) \triangleq \textstyle\sum_{c\in\mathcal{C}} |c| x_c \tag{10}$$

$$\text{s.t.} \quad \textstyle\sum_{c\in\mathcal{C}:e\in c} x_c \le u_e, \ \forall e \tag{11}$$

$$x_c \ge 0, \ \forall c \in \mathcal{C} \tag{12}$$

The set of Inequalities (11) ensures that the link capacity constraints are not violated. This problem can be viewed as a fractional cycle covering problem where the objective is to cover as much of the arc flows with cycles as possible. Instead of directly solving the linear programming problem, we use a fast primal-dual approximation algorithm to solve this linear program. The algorithm is very simple and can easily be implemented at the nodes. We first write the dual to the linear programming problem. Associating a dual variable l_e with the capacity constraint on arc e, we can write its dual as follows:

$$\min D(\mathbb{L}) \triangleq \textstyle\sum_e u_e l_e \tag{13}$$

$$\text{s.t.} \quad \textstyle\sum_{e\in c} l_e/|c| \ge 1, \ \forall c \in \mathcal{C} \tag{14}$$

$$l_e \ge 0, \ \forall e \tag{15}$$

The variables l_e can be viewed as the length of arc e. Following the FPTAS approach for the max-multicommodity flow problem in [20], we propose a primal-dual fast approximate algorithm for maximizing the amount of Phase 2 traffic eliminated at node i. However, unlike [20], the dual in our problem is not a shortest path problem but a minimum mean cycle problem which determines a cycle c with the minimum value of $\sum_{e\in c} l_e/|c|$. The minimum mean cycle problem can be solved in $O(\mathbb{V}\mathbb{E})$ time [21].

```
Data: Graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$; capacity $u_e$; approximation $\omega$;
Result: Maximum $F(\mathbb{X}) = \sum_c |c| x_c$; feasible flow
        allocation $\mathbb{Y} = [y_e]$ and path allocation $\mathbb{X} = [x_c]$;
Initialize $\epsilon := 1 - (1-\omega)^{1/2}$; $\delta := ((1+\epsilon)\mathbb{V})^{-(1-\epsilon)/\epsilon}$;
$l_e := \delta, \ \forall e$; $y_e := 0, \ \forall e$; $x_c := 0, \ \forall c$;
while dual objective $D(\mathbb{L}) < 1$ do
    Find Minimum-Mean-Cycle:
    $c^* :=$ minimum mean simple cycle in graph $\mathbb{G}$ under
    length system $\mathbb{L}$;
    Update Path:
    $\Delta x_{c^*} := \min_{e\in c^*} u_e$;
    $x_{c^*} := x_{c^*} + \Delta x_{c^*}$;
    $y_e := y_e + \Delta x_{c^*}, \ \forall e \in c^*$;
    $l_e := l_e(1 + \epsilon \Delta x_{c^*}/u_e), \ \forall e \in c^*$;
end
$x_c := x_c / \log_{1+\epsilon} 1/\delta, \ \forall c$;
$y_e := y_e / \log_{1+\epsilon} 1/\delta, \ \forall e$;
primal objective $F(\mathbb{X}) := \sum_c |c| x_c = \sum_e y_e$
```

Algorithm 1: Fast approximate algorithm

Theorem IV.2. *There exists a FPTAS that solves the maximum cycle canceling problem and obtains a solution within a factor of $(1-\omega)$ of the optimal solution in $O(\omega^{-2}\mathbb{V}\mathbb{E}^2 \ln \mathbb{V})$ time.*

Proof. Let $\bar{c}^*(\mathbb{L}) \triangleq \sum_{e\in c^*} l_e/|c^*| \triangleq \min_c \sum_{e\in c} l_e/|c|$ denote the mean of the minimum mean circle c^* under length system $\mathbb{L}$, then the dual problem is equivalent to finding a length system $\mathbb{L}$, such that $\beta \triangleq \min \frac{D(\mathbb{L})}{\bar{c}^*(\mathbb{L})}$ is minimized.

Consider the change of dual objective in each iteration,

$$\begin{aligned} D^{(i)} &= \sum_e u_e l_e^{(i)} = \sum_e u_e l_e^{(i-1)} + \epsilon \sum_{e\in c^{*(i-1)}} l_e^{(i-1)} \Delta x_{c^*}^{(i-1)} \\ &= D^{(i-1)} + \epsilon (F^{(i)} - F^{(i-1)}) \bar{c}^{*(i-1)}. \end{aligned} \tag{16}$$

Summing up all i,

$$D^{(i)} = D^{(0)} + \epsilon \sum_{j=1}^{i} (F^{(j)} - F^{(j-1)}) \bar{c}^{*(j-1)} \tag{17}$$

Since $\beta = \min \frac{D(\mathbb{L})}{\bar{c}^*(\mathbb{L})} \le \frac{D(\mathbb{L}^{(i)} - \mathbb{L}^{(0)})}{\bar{c}^*(\mathbb{L}^{(i)} - \mathbb{L}^{(0)})} = \frac{D^{(i)} - D^{(0)}}{\bar{c}^{*(i)} - \delta}$,

$$\bar{c}^{*(i)} \le \delta + \frac{\epsilon}{\beta} \sum_{j=1}^{i} (F^{(j)} - F^{(j-1)}) \bar{c}^{*(j-1)}. \tag{18}$$

Hence, $\bar{c}^{*(i)}$ is dominated by

$$\begin{aligned} g^{(i)} &= \delta + \frac{\epsilon}{\beta} \sum_{j=1}^{i} (F^{(j)} - F^{(j-1)}) g^{(j-1)} \\ &= g^{(i-1)} + \frac{\epsilon}{\beta} (F^{(i)} - F^{(i-1)}) g^{(i-1)} \\ &= g^{(i-1)} (1 + \epsilon (F^{(i)} - F^{(i-1)})/\beta) \\ &\le g^{(i-1)} e^{\epsilon (F^{(i)} - F^{(i-1)})/\beta} \\ &\le \delta e^{\epsilon F^{(i)}/\beta}. \end{aligned} \tag{19}$$

Suppose the process stops after t iterations, then

$$1 \leq \bar{c}^{*(t)} \leq \delta e^{\epsilon F^{(t)}/\beta}, \tag{20}$$

$$D(\mathbb{L}^{(t)}) = \beta \leq \frac{\epsilon F^{(t)}}{\ln(1/\delta)}. \tag{21}$$

We have set up the bound for the dual objective function in Inequality (21), yet we also need to bound the primal objective function so that an overall $(1-\omega)$-approximation can be guaranteed. For any edge e, every u_e units of flow routed through e increases l_e by at least $1+\epsilon$. On the other hand, $l_e \leq \mathbb{V}(1+\epsilon)$ since such an edge will never be increased any more through the iterations. The underlying reason is that any simple cycle c containing edge e will have $\sum_{e\in c} l_e/|c| \geq \mathbb{V}(1+\epsilon)/\mathbb{V} > 1$, which violates the stopping condition. So each edge capacity constraint in the primal problem is violated by at most $\log_{1+\epsilon}\frac{\mathbb{V}(1+\epsilon)}{\delta}$. By scaling the flows such that all capacity constraint are met, we have

$$F(\mathbb{X}^{(t)}) \geq \frac{F^{(t)}}{\log_{1+\epsilon}\frac{\mathbb{V}(1+\epsilon)}{\delta}}. \tag{22}$$

Thus the ratio of the dual and primal solutions can be bounded by

$$\frac{D(\mathbb{L}^{(t)})}{F(\mathbb{X}^{(t)})} \leq \frac{\epsilon \log_{1+\epsilon}\frac{\mathbb{V}(1+\epsilon)}{\delta}}{\ln(1/\delta)} \leq \frac{1}{1-\omega} \tag{23}$$

if $\epsilon = 1-(1-\omega)^{1/2}$ and $\delta = ((1+\epsilon)\mathbb{V})^{-(1-\epsilon)/\epsilon}$.

Applying the Weak Duality theorem, $F(\mathbb{X}^{(t)}) \leq \max F(\mathbb{X}) \leq D(\mathbb{L}^{(t)})$, hence the primal solution $F(\mathbb{X}^{(t)})$ is $(1-\omega)$-approximate.

In terms of complexity, because we saturate at least one edge in each iteration, and the total number of edge saturations is bounded by $\mathbb{E}\log_{1+\epsilon}\frac{\mathbb{V}(1+\epsilon)}{\delta}$, so the total number of iterations is also $\mathbb{E}\log_{1+\epsilon}\frac{\mathbb{V}(1+\epsilon)}{\delta}$ at most. Therefore, the overall time complexity is $O(\omega^{-2}\mathbb{V}\mathbb{E}^2\ln\mathbb{V})$. □

Each node can solve the maximum cycle canceling problem independently and hence can reduce the amount of Phase 2 traffic in the switch. Note that all these optimizations are done after the α_i values are picked by GLOBE-STS. These split ratios are not changed in GLOBE-DCC. Though it is easy to solve the problem for maximizing the flow on cycles, after many experiments, we see that almost the same benefit can be obtained by just making two-way swaps. As shown in Fig. 6, regardless of the number of nodes in the network, the average amount of reducible Phase 2 traffic is always about half of that in the traditional two-phase routing scheme, and the two arc cycle cancellation can eliminate most of the reducible traffic.

Considering that two-arc canceling is much easier to implement, it would be unnecessary to use maximum cycle canceling in most cases. The added advantage of using these short cycles is that it is easy to implement the scheme without any explicit traffic estimation using a simple counter based scheme. We call this variant GLOBE-ICC.

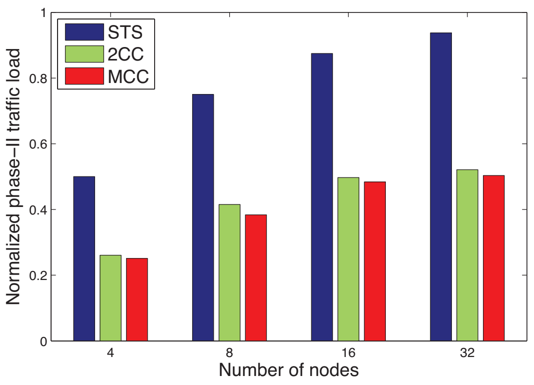

Fig. 6. Average traffic reduction of 2CC vs. MCC.

C. GLOBE-ICC: Implicit Two Arc Cycle Canceling Module

We now show that GLOBE-2CC can be implemented without explicit estimation of the rates or checking the conditions given in the last section. All rates are estimated implicitly and hence the name GLOBE-ICC. Each ingress node maintains $(n-1)$ Phase 1 VOQ denoted by $VOQ^1[i,j]$. These queues are used to store packets before Phase 1 is executed. Each ingress node also maintains $(n-1)$ Phase 2 VOQs denoted by $VOQ^2[i,j]$. These queues keep packets that have been routed to the node in Phase 1 and have to be routed to the packet's egress port in Phase 2. In addition each ingress node i maintains an $n\times n$ counter matrix. We refer to the counter matrix at node i as C^i and the entry in this matrix corresponding to node pair (j,k) as $C^i[j,k]$.

In each time slot each ingress node uses the statically configured sequence to determine the current matching egress port. Figure 7 shows the operations done at ingress node i for implementing GLOBE-ICC. At ingress node i, if a packet arrives for output node j, then we decrement the value of $C^i[j,k]$ by α_k and increment the value of $C^i[k,j]$ by α_j. When a packet is transmitted from $VOQ^1[i,m]$ to node j then the value of $C^i[m,j]$ is decremented by one. Also note that the end of any busy period represents a regeneration point and all counter are reset in order not to carry stale information. We show that this scheme implements GLOBE-2CC faithfully without any explicit rate estimation. Also note that all ingress ports make their decisions independently without any coordination. The only coordination is the computation of the α values by GLOBE-STS whenever the traffic matrix changes.

Theorem IV.3. *Given a static traffic matrix* $\mathbf{S}$*, if each ingress node implements the operations outlined in Figure 7 then the dynamic traffic matrix* $\mathbf{D}$ *that can be handled by the switch is given by the capacity region in Theorem IV.1*

Proof. (Outline)

From Step 1 in Figure 7, note that $C^i[j,k]$ tracks $\alpha_j d_{ik} - \alpha_k d_{ij}$. Consider the case in which ingress node i is matched

Operations at Ingress node i

1) Packet arrives externally for egress node j.
   - Set $C^i[j,k] \leftarrow C^i[j,k] - \alpha_k \quad \forall k \neq j$.
   - Set $C^i[k,j] \leftarrow C^i[k,j] + \alpha_j \quad \forall k \neq j$.
   - Add packet to $VOQ^1[i,j]$.
2) Let node i be matched to egress node j in this time slot.
   - If $VOQ^2[i,j] > 0$, transmit head of line packet to node j and decrement $VOQ^2[i,j]$.
   - If $VOQ^2[i,j] = 0$ then check if $VOQ^1[j] > 0$ and if it is then transmit the head of line packet from $VOQ^1[j]$ and decrement $VOQ^1[j]$.
3) If $VOQ^1[j] = 0$ then pick the queue m such that

$$m = \arg \max_{\{k: VOQ^1[i,k]>0\}} C^i[k,j].$$

4) Transmit the head of line packet from $VOQ^1[i,m]$ and decrement the value of $C^i[m,j]$ by one.
5) If all the queues at node i are empty then reset all counters $C^i[j,k] \leftarrow 0$.

Fig. 7. Description of Operations at Ingress node i

to egress node j but there are no packets (either Phase 1 or Phase 2) to transmit to j, then packets destined to some other destination node will be transmitted to j. In this case we identify a node k which has packets to transmit whose $C^i[k,j]$ value is the highest. Note that $C^i[k,j]$ tracks the value of $\alpha_k d_{ij} - \alpha_j d_{ik}$ and if this value is positive then we know that there will be Phase 2 traffic between nodes k and j due to traffic originating from node i. Hence a packet whose final destination is k is transmitted to node j as an intermediate node and the value of $C^i[k,j]$ is decremented. $\square$

We now conduct experiments to study the performance of all the algorithms.

V. Numerical Results

In this section, we present some numerical comparisons of GLOBE-STS, GLOBE-2CC, and GLOBE-ICC. GLOBE-STS only runs the first module of GLOBE, while GLOBE-2CC and GLOBE-ICC also run DCC modules in addition to the STS module. All experiments were performed on a 100 node network. First, we evaluate the performances of three different schemes under a hot-spot Bernoulli i.i.d. arrival process with

$$t_{ij} = \begin{cases} 0.5, & \text{if } j = i+1 \\ 0, & \text{if } j = i \\ 0.5/(N-2), & \text{otherwise.} \end{cases} \tag{24}$$

Note that all traffic is treated as dynamic in this experiment.

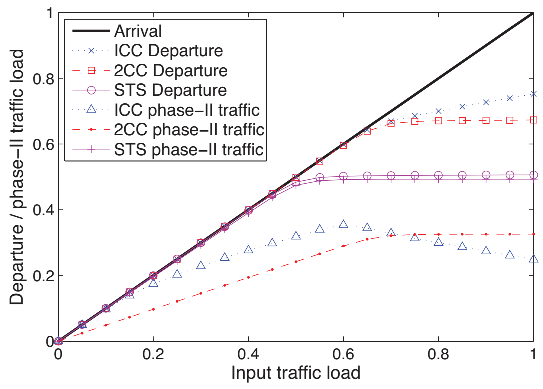

Fig. 8. Throughput performances of different schemes in a 100 node network under dynamic hot-spot Bernoulli i.i.d. traffic.

GLOBE-STS sends traffic uniformly and randomly to different intermediate nodes, regardless of the final destinations. GLOBE-2CC leverages prior knowledge of the rate matrix, and eliminates unnecessary Phase 2 traffic via two cycle canceling. In both cases, a Phase 1 packet from input i to output k is classified to be relayed via j upon arrival, and will be sent to j whenever there is no Phase 2 packets waiting to be delivered to the same destination. The last scheme is GLOBE-ICC, which implements the algorithm in Figure 7.

GLOBE-2CC with rate-aware two cycle canceling extends the stability region of GLOBE-STS by one-third in this case, as shown in Fig. 8. Such advantage is achieved by supressing unnecessary Phase 2 traffic. GLOBE-ICC scheme performs even better. The underlying reason is that GLOBE-ICC not only suppresses unnecessary Phase 2 traffic according to the two-way swap heuristics, but also prioritizes one-hop delivery, be it Phase 1 or Phase 2. Hence when the incoming traffic is overloading, GLOBE-ICC is more likely to send one-hop packets and refrains from generating two-phase traffic.

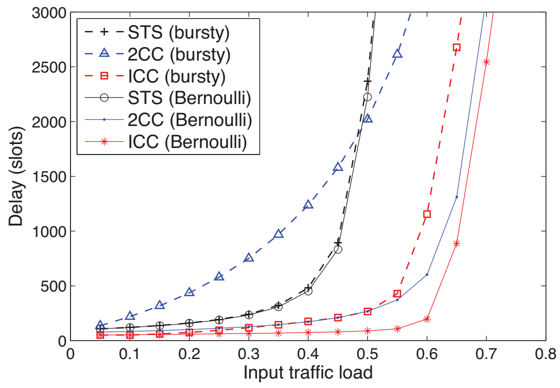

Fig. 9. Delay performances of different schemes in a 100 node network under dynamic hot-spot Bernoulli i.i.d. input traffic.

Fig. 9 plots the delay performances in this experiment.

Under Bernoulli i.i.d. traffic, GLOBE-2CC not only extends the stability region, but also reduces the average delay. GLOBE-ICC does a even better job in both aspects. However, under bursty traffic, GLOBE-2CC with prior knowledge of the long term traffic pattern works very badly. Even though the stability region should not change, the average delay increases dramatically. Such a worsened performance is due to the high burstiness of the incoming traffic and the large fluctuations in the realtime traffic matrix. In such cases, GLOBE-2CC should be accompanied with a realtime rate estimation mechanism, but that is out of the scope of this paper. To the contrary, the performance of GLOBE-STS is not affected since it is rate-oblivious. GLOBE-ICC also performs worse in this case because of the rapidly changing traffic matrices, but to a much smaller extent.

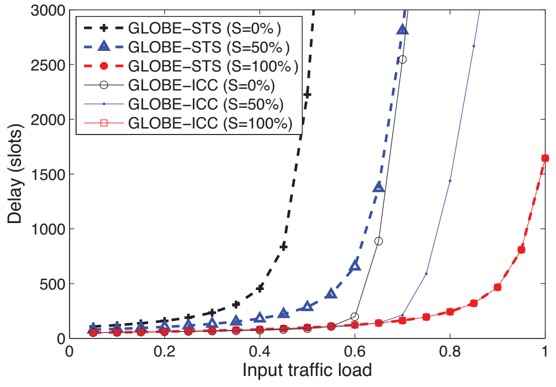

Fig. 10. Performance of GLOBE under static and dynamic traffic.

Finally, the effect of the GLOBE scheme on the overall throughput and delay performances is studied in a 100 node interconnection network with both static and dynamic traffic. The static traffic is uniform Bernoulli i.i.d., and hence is always sent directly to the destinations (according to the result of BV decomposition). The dynamic traffic is hot-spot Bernoulli i.i.d., and is taken care of by STS and ICC modules of GLOBE. As shown in Figure 10, at one end, when there is no static traffic, GLOBE has no effect, so only 50% of the traffic is sustainable by STS, and about 70% throughput is achived by ICC. At the other end, when all traffic is static and known, GLOBE-STS supports 100% throughput through BV decomposition. The delay performance also improves as the proportion of static traffic increases.

VI. Conclusion

In this paper, we address the data center interconnection design problem, and propose a novel generalized load balanced scheduler GLOBE, that handles both static and dynamic components in data center traffic. GLOBE bridges a Birkhoff-von Neumann scheduler that is suitable for known static traffic with a load balanced scheduler that offers best-effort service to dynamic traffic. The two schedulers are optimally combined into one single system, and complement one another. An opportunistic distributed cycle canceling module DCC is also designed to further reduce the unnecessary traffic due to load balancing and improve the performance. As demonstrated through theoretical analyses and numerical simulations, GLOBE-STS guarantees 100% throughput for static traffic and optimally schedules dynamic traffic for the worst case, while GLOBE-DCC further enlarges the capacity region and reduces the average delay.

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 for Modular Data Centers,” *ACM SIGCOMM*, 2010.
[2] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, T.S. Eugene Ng, M. Kozuch, and M. Ryan, “c-Through: Part-time Optics in the Data Center,” *Sigcomm*, 2010.
[3] C. Kachris, and I.Tomkos, “A Survey on Optical Interconnects for Data Centers,” *IEEE Communications Surveys and Tutorials*, Fourth Quarter, 2012.
[4] G. Porter, R. Strong, N. Farrington, A. Forencich, P.C. Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, “Integrating Microsecond Circuit Switching into the Data Center,” *Sigcomm*, 2004.
[5] S. Kamil, D. Gunter, M. Lijewski, L. Oliker, and J. Shalf, “Reconfiguring Hybrid Interconnections for Static and Dynamic Scientific Applications,” *Proc. Conference on Computing Frontiers*, 2007.
[6] S. Kandula, J. Padhye, and P. Bahl, “Flyways to De-congest Data Center Networks,” *ACM-Hotnets VIII*, October 2009.
[7] T.A. Benson, A. Anand, A. Akella, and M. Zhang, “Understanding Data Center Traffic Characteristics,” *Proceedings of Workshop: Research on Enterprise Networking*, August 2009.
[8] C. Minkenberg, F. Abel, P. Muller, R. Krishnamurthy, M. Gusat, P. Dill, I. Iliadis, R. Luijten, R.R. Hemenway, R. Grzybowsky, and E. Schiattarella, “Designing a Crossbar Scheduler for HPC Applications,” *IEEE Micro*, Vol. 26, No. 3, 2006.
[9] O. Liboiron-Ladouceur, A. Shacham, A.B. Small, B.G. Lee, H. Wang, C.P. Lai, A. Biberman, and K. Bergman, “The Data Vortex Optical Packet Switched Interconnection Network,” *Journal of Lightwave Technology*, Vol. 26, No. 13, 2008.
[10] X. Ye, Y. Yin, S.J.B. Yoo, P. Mejia, R. Proietti, and V. Akella, “DOS: a scalable optical switch for datacenters,” *ANCS*, 2010.
[11] N. Mckeown, A.Mekkittikul, V.Ananthram and J.Walrand, “Achieving 100% Throughput in an Input Queued Switch,” *IEEE Transactions on Communications*, 1999.
[12] C.S. Chang, D.S. Lee, and Y.S. Jou, “Load Balanced Birkhoff-vonNeumann Switches: Part I: One Phase Buffering,” *Computer Communications*, 2002.
[13] S. Kandula, D. Katabi, S. Sinha, and A. Berger, “Dynamic Load Balancing without Packet Reordering,” *ACM Computer Communication Review*, Vol. 37, No. 2, 2007.
[14] I. Keslassy, S.T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, “Scaling Internet Routers using Optics,” *Sigcomm*, 2004.
[15] I. Keslassy, “Load Balanced Router,” Ph.D. Dissertation, Stanford University, 2004.
[16] I. Keslassy, and N. Mckeown, “Maintaining Packet Order in Two Phase Switches,” *IEEE INFOCOM*, 2002.
[17] B. Mukherjee, “WDM Optical Communication Networks: Progress and Challenges,” *IEEE JSAC*, October 2000.
[18] A. Goel, M. Kapralov, and S. Khannal, “Perfect Matchings in $O(n \log n)$ in Regular Bipartite Graphs,” *ACM-STOC*, 2010.
[19] H.Liu and R. Zhang-Shen, “On Direct Routing in the Valiant Load-Balancing Architecture,” *IEEE-GLOBECOMM*, November 2005.
[20] N. Garg, and J. Konemann, “Faster and simpler algorithms for multi-commodity flow and other fractional packing problems,” SIAM Journal on Computing, 37(2):630-652, 2007.
[21] R.M. Karp, “A characterization of the minimum cycle mean in a digraph,” Discrete Mathematics, 23(3):309-311, 1978.