Received June 24, 2020, accepted July 5, 2020, date of publication July 8, 2020, date of current version July 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007954

A Partial Store-and-Forward Scheduling Method for Inter-Datacenter Bulk Data Transfers

XIAO LIN[1], (Member, IEEE), SHENGNAN YUE[2], (Graduate Student Member, IEEE), YUANLONG TAN[3], WEIQIANG SUN[2,4], (Senior Member, IEEE), MALATHI VEERARAGHAVAN[3], (Fellow, IEEE), AND WEISHENG HU[2], (Senior Member, IEEE)

[1]College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China
[2]State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China
[3]*(Deceased)* The Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA
[4]Shanghai Institute for Advanced Communication and Data Science, Shanghai 200240, China

Corresponding author: Xiao Lin (linxiaocer@fzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901118 and Grant 61433009, and in part by the Open Foundation of the State Key Laboratory of Advanced Optical Communication Systems and Networks under Grant 2019GZKF03003.

ABSTRACT Temporarily storing delay-tolerant data at peak hours and forwarding the data at off-peak hours, i.e., performing Store-and-Forward (SnF) using datacenter storage, can mitigate peak-hour congestion and exploit off-peak-hour bandwidth in inter-datacenter networks. Most prior studies considered a case where all nodes along their routing paths provided SnF options for the scheduling decision making. Intuitively, their solutions maximize the scheduling flexibility. However, the computational complexity of their solutions increases exponentially with the hop count. Meanwhile, SnF operations are generally performed on a portion of nodes rather than every node along the path. Thus, the considered case seems to be unnecessary in practice. In this paper, our studies reveal that desirable performance can be attained by involving a portion of nodes rather than all nodes along the path in the decision making. Thus, we propose a partial SnF scheduling method, which involves a portion of nodes in scheduling and introduces a network abstraction based on the involved nodes. Simulations demonstrate that the proposed method has lower complexity while achieving high performance.

INDEX TERMS Bulk data transfers, inter-datacenter networks, wavelength routing, storage.

I. INTRODUCTION

A. BACKGROUND

Emerging online services have brought a rapid growth in bulk transfers across geo-distributed datacenters (DCs) [1]. The bandwidth-hungry and long-lived nature of bulk data flows imposes a great challenge on the inter-datacenter networks (inter-DCNs) [2]–[4]. To tackle this, prior studies exploited the delay-tolerance of bulk transfers to provide extra flexibility in scheduling and resource allocation [2]–[7].

A key feature of the prior studies is that they delivered bulk data over end-to-end (E2E) connections. Due to the diurnal traffic pattern, the bursty user demands and the time zones between DC sites, the network traffic can peak on different sites and at different hours. As a result, residual bandwidth in inter-DCNs varies in both time and space [7]. The bandwidth bottleneck of a bulk data flow may occur on different links and at different hours. Thus, it is difficult to fully utilize the residual bandwidth over E2E connections [8]. To accommodate the growing peak-hour demands, DC operators have to constantly purchase bandwidth from ISPs or upgrade the capacities of their dedicated lines even if large amounts of bandwidth are left unused at the off-peak hours.

An alternative solution is to introduce the storage of DC sites into data-plane paths. The intermediate DC sites can temporarily store delay-tolerant data at the peak hours and forward the data at the off-peak hours. On the one hand, the peak-hour congestion is mitigated by shifting the data transmissions to off-peak hours. On the other hand, the off-peak-hour residual bandwidth can be exploited and used for bulk data transfers. This solution is called store-and-forward (SnF) and has proven to be effective in inter-DCNs [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Rentao Gu.

In this paper, we consider a combination of DC storage and optical circuit switching (OCS) for inter-DC bulk transfers. On the one hand, OCS enables high-bandwidth paths for bulk data with minimum control and processing overhead. On the other hand, E2E constraints are relaxed in the presence of SnF. This greatly improves utilization when compared to conventional OCS [9]–[13]. However, the use of storage transforms the conventional routing problem into a complex scheduling problem, where both bandwidth and storage resources must be allocated and both spatial routing and temporal scheduling must be performed [11]. Solving such a scheduling problem is computationally expensive [8].

B. MOTIVATION

In essence, the flexibility of an SnF solution depends on the number of storage nodes along its routing path. Each storage node provides an SnF option for the scheduling decision making. The more storage nodes, the more flexible the SnF solution, and the more residual bandwidth can be utilized for bulk data transfers. Most prior studies aimed to fully leverage the flexibility and hence considered a case where all nodes along the routing path were involved in their scheduling problems [8]–[19]. Upon receiving data, each node has to decide whether the data should be stored, for how long, and at what rate the data should be transmitted to the next "hop" along the path. The computational complexity of the scheduling problem increases exponentially with the hop count [8], [13]. Thus, the problem may become computationally intractable for large networks and dynamic traffic.

In practice, SnF operations are often performed on a portion of the nodes rather than every node along its routing path in each bulk transfer [11], [14]. For example, the reported work demonstrated that the majority of requests could reach their destinations through one or two SnF operations [14]. Moreover, in the context of inter-DC optical networks, each SnF operation will require an expensive OEO conversion, which consumes extra power and introduces control overhead [9]. This naturally inspires us to explore how to involve a portion of nodes rather than all nodes along the routing path in the decision making, thereby reducing the computational complexity.

C. CONTRIBUTIONS

Our contributions are summarized as follows:

1) We first present analytic models to compare SnF with two typical E2E provisioning approaches in terms of performance and complexity. Our key finding shows that desirable performance can be attained by using a storage node rather than multiple storage nodes in scheduling under certain circumstances.
2) We further extend the analytic models to quantify the impact of the number of storage nodes along the routing path on the performance and the complexity of SnF. Studies show that involving a portion of nodes in scheduling while expanding the time horizon of temporal scheduling can provide more performance benefits while imposing less computational burden.
3) Inspired by this finding, we propose a partial SnF (pSnF) scheduling method. On the one hand, the pSnF method only involves a portion of nodes along the path in the scheduling decision making. On the other hand, the pSnF method provides an abstract view of the future network resources based on the involved nodes. Given the limited computational cost, the pSnF method can search network state further ahead in time and hence make more optimal decisions than the conventional method using all nodes in scheduling. Simulations demonstrate that the pSnF method can outperform the conventional method in terms of blocking probability and computing time.

The rest of the paper is organized as follows. Sect. II reviews the literature. Sect. III compares the three approaches. Sect. III studies the impact of the number of storage nodes. Sect. V presents the pSnF method, which is followed by evaluations and discussions in Sect. VI. Sect. VII concludes this paper.

II. RELATED WORK

A. SnF SCHEDULING METHODS

We classify the existing SnF scheduling methods into two categories: (1) full SnF schemes, where all network nodes were involved in the scheduling decision making; (2) partial SnF schemes, where only a portion of network nodes were involved in the decision making. These efforts are summarized in Fig. 1.

1) FULL SnF SCHEMES

Most prior studies attempted to fully leverage the network infrastructure and maximize the flexibility of scheduling. They considered the scenarios where storage was deployed on all network nodes. Meanwhile, all nodes along a routing path were involved in the decision making [8]–[19]. The SnF scheduling problems were formulated as optimization problems, such as linear programming problems and network flow problems [8]–[11], [16]–[19]. Classical optimization algorithms or heuristics were used to achieve optimal solutions in routing, scheduling, and resource allocation. The prior proposed methods have proven to be effective for small networks or static traffic where request arrival time is fixed and given in advance. However, the SnF scheduling problem has been proven to be NP-hard [8]. Our prior studies revealed that with the full SnF scheme, the size of the SnF scheduling problem exponentially increased with the hop count of the routing path [12]. This implies that the problem may become intractable when the problem size is large. Thus, the prior methods may be too complex to implement for large networks and dynamic traffic where requests arrive one after another randomly.

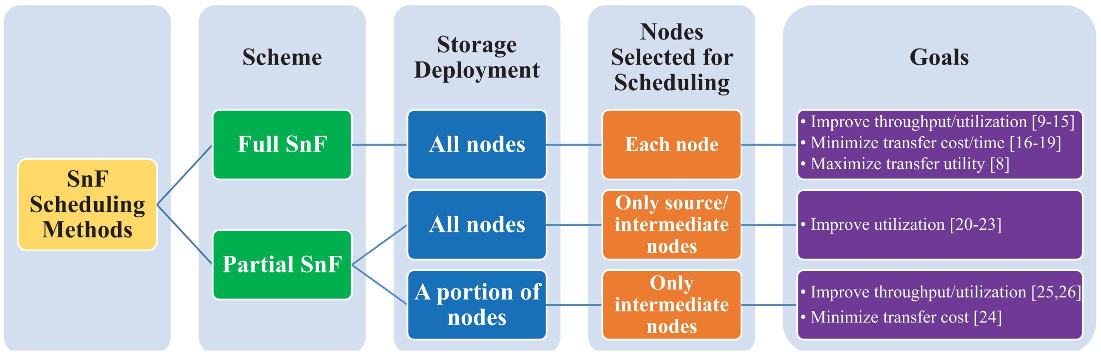
Scheme
Storage Deployment
Nodes Selected for Scheduling
Goals
SnF Scheduling Methods
Full SnF
Partial SnF
All nodes
All nodes
A portion of nodes
Each node
Only source/ intermediate nodes
Only intermediate nodes
• Improve throughput/utilization [9-15]
• Minimize transfer cost/time [16-19]
• Maximize transfer utility [8]
• Improve utilization [20-23]
• Improve throughput/utilization [25,26]
• Minimize transfer cost [24]

FIGURE 1. **Summary of SnF scheduling methods.**

2) PARTIAL SnF SCHEMES

Some reported studies also considered the scenarios where storage was deployed on all network nodes [20]–[23]. However, they imposed some constraints on scheduling. Feng *et al*. preferably delivered data through the minimum number of SnF operations [20]. Their proposed scheduling method used a greedy algorithm to find a viable solution starting from one SnF operation allowed. However, the greedy algorithm failed to gain practical insights into the impact of the number of storage nodes on the performance and the complexity of SnF. To reduce the high-speed read/write burden on the storage systems, the authors in [21] assumed that only sources could store data. Patel *et al*. compared SnF with Advance Reservation (AR) [22], [23]. AR was considered as a special case of SnF when only the source was SnF-enabled. Their studies showed that compared to SnF, the performance benefits provided by AR were slight, especially when the traffic load was high.

Other reported studies considered the scenarios where storage was deployed on a portion of network nodes [24]–[26]. In [25], when E2E lightpath was unavailable for a particular request, the proposed scheduling method attempted to establish a lightpath from the source to the closest storage node for temporarily storing, and then established a lightpath from this storage node to the destination at a later time. Although a routing path may traverse multiple intermediate nodes, only an intermediate node was SnF-enabled, which resulted in a limited performance improvement provided by SnF. Moreover, the authors formulated the storage location problem as a facility location problem and solved it by graph centrality measures. The storage deployment mainly depended on the characteristics of network topology, but without considering any complexity issues. Iosifidis *et al.* aimed to maximize the network data transfer capability with the minimum required storage capacity [26]. They formulated the joint optimization problem of routing and storage usage as max-flow problems. However, they did not consider how the number of storage nodes affected the complexity of SnF. Laoutaris *et al*. considered a 3-node tandem network, where only the intermediate node was SnF-enabled [24]. They developed an analytical model for transferring bulk data through single-hop and single-path transfers but did not consider any routing or scheduling issues. None of the aforementioned literature investigated the impact of the number of storage nodes on the complexity and the performance of SnF.

3) LESSONS LEARNT FROM PRIOR WORK

In short, most prior studies considered the full SnF schemes. Although their proposed scheduling methods have proven to be effective for small networks or static traffic, they may become too complex to implement for large networks and dynamic traffic. By contrast, some proposed scheduling methods considered the partial SnF schemes by only using either source or intermediate nodes for scheduling. However, they aimed to minimize storage usage or deployment, without considering any complexity issues. There exists a tradeoff between performance and complexity in the design of an SnF scheduling method. An efficient SnF scheduling method should carefully examine the number of storage nodes involved in the decision making, but little research has considered this.

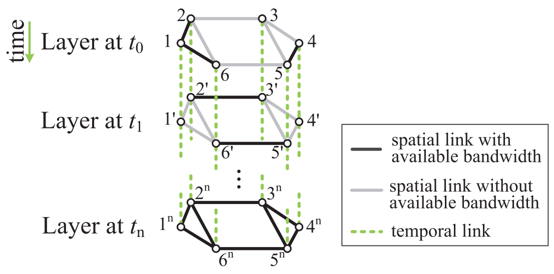

FIGURE 2. **A time-shifted multilayer graph (TS-MLG).**

B. TIME-SHIFTED MULTILAYER GRAPH

TS-MLG is a routing framework proposed for bulk transfers in OCS network with SnF [14]. Fig. 2 shows an example of TS-MLG, which consists of a set of layers. Each layer is a snapshot of the network state at a certain instant. The layers are stacked downward in a time-increasing order to capture the dynamics of resource usage. For example, the topmost layer indicates that link 6-5 is unavailable at time t_0.

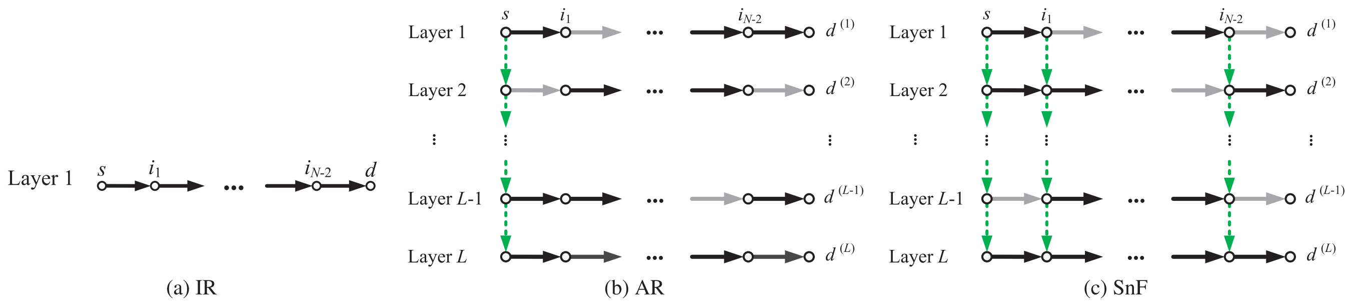

FIGURE 3. **Routing models (i.e., TS-MLGs) of IR, AR and SnF.**

The second topmost layer indicates that link 6-5 becomes available at time t_1. Besides, spatial links connect different nodes within the same layer, and temporal links connect the same nodes at different layers. When a request traverses a spatial link, it suggests data transfer from one node to another. When a request traverses a temporal link, it suggests data storage on a node for a certain period. The capacity of a spatial link or a temporal link refers to the bandwidth of that link, or the storage capacity of that node. For simplicity, the link capacity in Fig. 2 is assumed to be one wavelength per link. In practice, the link capacity is usually larger than one.

Consider a request r from node 1 to node 5 arrives at the network at t_0. An E2E transmission is unavailable at t_0. However, by performing a routing algorithm, e.g., Dijkstra's algorithm, on the TS-MLG shown in Fig. 2(a), an E2E path over time and space is found, i.e., path 1-6-6'-5'. In this case, SnF operation is performed on node 6. With the TS-MLG, an SnF scheduling problem is formulated as a routing problem.

The arrival and departure of requests change the network states, thus update the TS-MLG dynamically. New layers are added to or expired layers are removed from the TS-MLG dynamically. Consider the aforementioned request r. Once r is admitted, the link states of link 1-6 and link 6'-5' in the two topmost layers will be updated to unavailable. Then, two new layers will be added to the TS-MLG, with each layer corresponding to the moment when the transmission of r completes for link 1-6 and link 6'-5' (i.e., their link states on the new layers are available), respectively. As time progresses, the topmost layer will be removed from the TS-MLG when the current moment corresponds to the next layer, i.e., the second topmost layer.

The computational complexity of a routing algorithm depends on the size of a network graph. The TS-MLG size is equal to the number of layers times the number of nodes in the network topology. Thus, the number of layers dominantly determines the complexity. The bursty nature of the bulk data flows may result in a temporary increase in the number of layers. This suggests high computational complexity on the route search for new requests. In other words, the computational complexity may vary request-by-request. To bound the complexity, the number of layers used for routing needs to be limited. As a result, requests are accepted or blocked depending on whether viable paths are found within a given number of layers.

III. COMPARISONS OF IR, AR AND SnF

Immediate Reservation (IR) and Advance Reservation (AR) [27] are two typical E2E provisioning approaches in OCS networks. IR and AR can be regarded as special cases of SnF when no node or only source is SnF-enabled. To understand their pros and cons, we first briefly review the rationales of the three provisioning methods. Then, we present the analytic models and compare SnF with IR and AR in terms of performance and complexity.

In **IR**, a request is accepted or rejected, depending on whether the bandwidth is available for an E2E transfer. In other words, upon arrival, a decision is made immediately according to the current network state, i.e., bandwidth availability information at the moment of arrival. In **AR**, the transfer of a request can be deferred to a later time when an E2E connection is available. Such a deferral can be performed by only the source of each request. Thus, a decision is made according to both the current and the future network states, i.e., the bandwidth availability information over a certain period. Here, the full SnF scheme is considered. Thus, in **SnF**, each node can perform SnF. A decision is made according to both the current and the future states of both bandwidth and storage availability information.

A. ANALYTIC MODELS

We extend the prior analytic models [12] to provide a comparison between IR, AR and SnF. For simplicity, consider a fixed route $R = \{s, i_1, \ldots, i_{N-2}, d\}$, where s is the source, d is the destination, N denotes the number of nodes on R and $N \geq 2$. Routing models (i.e., TS-MLGs) of IR, AR and SnF are depicted in Fig. 3. Assume that the L topmost layers can be used for scheduling.

The TS-MLG of IR only consists of a layer, as depicted in Fig. 3(a). This is because only the current network state is considered in IR. In Fig. 3(b), the TS-MLG of AR consists of L layers, since both the current and the future network states are considered in AR. Additionally, only the sources at different layers are connected via temporal links.

In Fig. 3(c), since each node can perform SnF, the TS-MLG consists of multiple layers and temporal links. Let $d^{(l)}$ denote the destination d at layer l in the TS-MLG, where $l \in [1, L]$. These destinations are not connected via temporal links, because an alternate path should not traverse multiple destinations.

In this paper we differentiate between "path" and "route." We use the common term "route" to refer to a path in a conventional network graph, which simply shows how to reach a destination in the spatial dimension, without considering the temporal dimension. In a TS-MLG, a path traverses multiple spatial and temporal links, which suggests that the SnF operations are performed on different nodes and at different time. An alternate path is defined as a path from s to $d^{(l)}$ in the TS-MLG, without considering the resource availability of each link, where $l \in [1, L]$. A viable path is defined as an alternate path with the required bandwidth and storage resources on each link.

With the TS-MLG, the scheduling problem can be formulated as the routing problem. A simple solution to the routing problem is to list all alternate paths from s to d and search for a viable path. Thus, the number of alternate paths is a native measure of the computational complexity of the problem. For instance, there is only one alternate path for the node pair (s, d) in Fig. 3(a). When the layers are introduced in the model, an alternate path can reach the destinations at different layers. In Figs. 3(b) and (c), the number of alternate paths for (s, d) increases with the number of layers. Here, $P_{(s,d)}(N, L)$ is defined as the total number of alternate paths from s to $d^{(\boldsymbol{L})}$, where $\boldsymbol{L} = [1, \ldots, L]$. Here, $P_{(s,d)}(N, L)$ is used to quantify the computational complexity.

In addition, the main idea of SnF is to trade storage for bandwidth. Thus, the probability of resources being available should be considered in the analytic models. Probability of reservation failure $F_{(s,d)}(N, L)$ is defined as the probability that a request fails to find and reserve any viable path from a set of alternate paths of size $P_{(s,d)}(N, L)$. Let p_b and p_s denote the probability that a request fails to reserve the required bandwidth on a spatial link, and the required storage on a temporal link, respectively. $F_{(s,d)}(N, L)$ is used to quantify the achievable performance of the problem. Table 1 lists $P_{(s,d)}(N, L)$ and $F_{(s,d)}(N, L)$ in the cases of IR, AR and SnF.

TABLE 1. $P_{(s,d)}(N, L)$ **and** $F_{(s,d)}(N, L)$ **in IR, AR and SnF.**

	IR	AR	SnF
Number of alternate paths	1	L	$P^{SnF}_{(s,d)}(N, L)$
Probability of reservation failure	$F^{IR}_{(s,d)}(N)$	$F^{AR}_{(s,d)}(N, L)$	$F^{SnF}_{(s,d)}(N, L)$

In Fig. 3(a), there exists one layer in the TS-MLG of IR. Thus, $P^{IR}_{(s,d)} = 1$. As N-1 spatial links are connected in series, a request will be blocked if any one of the N-1 spatial links cannot provide the required bandwidth. Thus, we have

$$F^{IR}_{(s,d)}(N) = 1 - (1 - p_b)^{N-1} \tag{1}$$

In Fig. 3(b), there exist L layers in the TS-MLG of AR. Thus, $P^{AR}_{(s,d)}(L) = L$. Since the L alternate paths share some common temporal links, their reservation failure probabilities should be dependent. For simplicity, we assume that temporal links along each alternate path are independent. A request will be blocked only when all the L alternate paths are unavailable. We have

$$F^{AR}_{(s,d)}(N, L) = \prod_{l=1}^{L} [1 - (1 - p_s)^{l-1}(1 - p_b)^{N-1}] \tag{2}$$

The expressions of $P^{SnF}_{(s,d)}(N, L)$ and $F^{SnF}_{(s,d)}(N, L)$ have been illustrated in [12]. Here, we present

$$P^{SnF}_{(s,d)}(N, L) = \begin{cases} \sum_{l=1}^{L} P^{SnF}_{(i_1,d)}(N-1, l) & \text{if } N \geq 3 \\ L & \text{if } N = 2 \end{cases} \tag{3}$$

$$F^{SnF}_{(s,d)}(N, L) = \begin{cases} \prod_{l=1}^{L} [1 - (1 - p_s)^{L-l}(1 - p_b) \\ \times(1 - F^{SnF}_{(i_1,d)}(N-1, l))] & \text{if } N \geq 3 \\ \prod_{l=1}^{L} [1 - (1 - p_s)^{l-1}(1 - p_b)] & \text{if } N = 2 \end{cases} \tag{4}$$

N and L determine the TS-MLG size. Increasing N suggests selecting a longer-spatial-hop route for requests, while increasing L suggests expanding the time horizon of temporal scheduling, e.g., enabling a longer storing time.

B. EVALUATION OF PERFORMANCE AND COMPLEXITY

We evaluate the properties of $P_{(s,d)}(N, L)$ and $F_{(s,d)}(N, L)$ in the cases of IR, AR and SnF. Numerical results are shown in Fig. 4.

Figs. 4(a) and (b) compare $P_{(s,d)}(N, L)$ in IR, AR and SnF. $P^{IR}_{(s,d)}$ remains 1 regardless of N and L. $P^{AR}_{(s,d)}$ linearly increases with L, but regardless of N. Thus, the results of IR and AR in Fig. 4(a) and (b) are the same. By contrast, $P^{SnF}_{(s,d)}(N, L)$ increases with L. The larger value of N, the more significant increase in $P^{SnF}_{(s,d)}(N, L)$. Compared to IR and AR, SnF enables more alternate paths, which suggests that requests are more likely to find viable paths in SnF. However, a larger set of alternate paths also imposes more computational burden on searching.

We assume $p_s \ll p_b$ in Fig. 4(c) because storage is more sufficient than bandwidth in typical inter-DCNs. However, in some metro scenarios, such as the Central Office Re-architected as a Datacenter (CORD) and micro DCs for edge computing, the storage resources may not be sufficient to handle all bulk data flows. The values of p_b and p_s in such scenarios are worth of further study.

Fig. 4(c) shows that $F_{(s,d)}(N, L)$ in the three cases increases with N. It is difficult to find viable paths along a long-spatial-hop route (larger value of N). $F^{SnF}_{(s,d)}(N, L)$ is lower than the others. But the gap between $F^{AR}_{(s,d)}(N, L)$ and $F^{SnF}_{(s,d)}(N, L)$

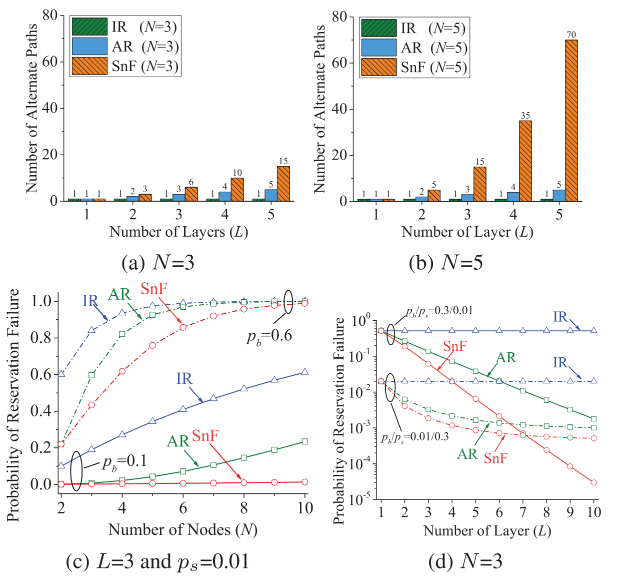

FIGURE 4. Number of alternate paths (a) and (b) and probability of reservation failure (c) and (d) in IR, AR and SnF.

remains small when N=2∼6 and p_b=0.1. The gap widens with N. With p_b increasing, it becomes difficult to reserve the required bandwidth. This results in a great increase in $F_{(s,d)}(N, L)$ in the three cases.

In Fig. 4(d), $F^{AR}_{(s,d)}(N, L)$ and $F^{SnF}_{(s,d)}(N, L)$ decrease with L, but $F^{IR}_{(s,d)}(N)$ remains constant, when p_b/p_s=0.3/0.01. The gap between $F^{AR}_{(s,d)}(N, L)$ and $F^{SnF}_{(s,d)}(N, L)$ widens with L. Requests can search more network resources further ahead in time with L increasing. The aforementioned studies assume $p_s \ll p_b$. Without loss of generality, we further assume $p_s \gg p_b$. Herein, p_b/p_s=0.01/0.3. This means that the bandwidth is sufficient but the storage is scarce. In this case, increasing L becomes less effective in reducing $F^{AR}_{(s,d)}(N, L)$ and $F^{SnF}_{(s,d)}(N, L)$.

C. LESSONS LEARNED FROM ANALYSIS

In a typical inter-DCN where storage is sufficient, our key findings are as follows:

1) While IR and AR obtain low complexity, their performance may be insufficient for large networks and dynamic traffic. SnF provides performance advantages over IR and AR, at the cost of higher complexity.
2) The performance gap between AR and SnF remains small when N, L and p_b is small. This suggests that desirable performance can be attained with a storage node rather than multiple storage nodes under such circumstance. But this gap greatly widens with L increasing. Thus, SnF has more significant performance advantages over AR with a larger time horizon of the temporal scheduling.
3) In essence, the high performance and the high complexity associated with SnF are due to the fact that SnF enables more storage nodes along routing paths when compared to IR and AR. Thus, it is worthwhile to study how the number of storage nodes affects SnF.

IV. IMPACT OF THE NUMBER OF STORAGE NODES

In this section, we first extend the analytic models to quantify the impact of the number of storage nodes on SnF. Then, we compare the partial SnF scheme with the full SnF scheme.

A. ANALYTIC MODELS

Consider a fixed route $R = \{s, i_1, i_2, \ldots, i_{N-2}, d\}$. Let N_s denote the number of storage nodes on R, where $N_s \in [1, N-1]$. The destination should not be used for temporary storage. Fig. 5 shows an example routing model of the partial SnF scheme, i.e., the partial model, where the first N_s nodes on R are storage nodes and L_s layers can be used for scheduling. Note that L and L_s denote the number of layers used for the full SnF scheme and the partial SnF scheme, respectively. L_s is independent of L.

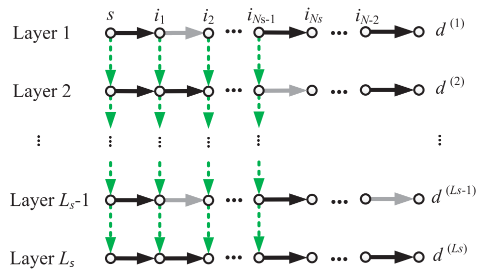

FIGURE 5. Routing model of the partial SnF scheme with N_s storage nodes and L_s layers.

When $N_s = 1$ and the source is the storage node, the partial model is equivalent to the AR model shown in Fig. 3(b). When $N_s = N - 1$, the partial model is equivalent to the SnF model shown in Fig. 3(c), i.e., the full model. When $1 < N_s < N - 1$, there exist multiple storage deployment schemes.

$P^{SnF}_{(s,d)}(N, N_s, L_s)$ is defined as the total number of alternate paths from s to $d^{(\boldsymbol{L_s})}$, where $\boldsymbol{L_s} = [1, 2, \ldots, L_s]$. $P^{SnF}_{(s,d)}(N, N_s, L_s)$ is independent of the storage deployment schemes. This is because only storage nodes are connected via temporal links. Routing options for each storage node are either its downstream neighbor (i.e., next spatial hop) or itself in the future (i.e., next temporal hop); while routing option for each non-storage node is confined to its downstream neighbor. Non-storage nodes cannot diverge various alternate paths. The partial model hence is equivalent to a smaller full model. Specially, a partial model with N nodes on a route, N_s storage nodes and L_s layers is equivalent to a full model with N_s+1 nodes on a route and L_s layers. Thus, we have

$$P^{SnF}_{(s,d)}(N, N_s, L_s) = P^{SnF}_{(s,d)}(N_s + 1, L_s) \quad (5)$$

Given a destination $d^{(l)}$, all the alternate paths from s to $d^{(l)}$ have the same spatial hop count (i.e., $N-1$) as well as temporal hop count (i.e., $l-1$), regardless of particular storage deployment schemes. Since p_b and p_s are assumed to be independent from each link, these alternate paths have the same probability of reservation failure, regardless of the storage deployment schemes as well. In other words, the probability of reservation failure is determined by N, N_s and L_s, and independent of the storage deployment schemes.

$F^{SnF}_{(s,d)}(N, N_s, L_s)$ is defined as the probability that a request fails to find and reserve any viable path from a set of alternate paths of size $P^{SnF}_{(s,d)}(N, N_s, L_s)$. For simplicity, the storage deployment scheme shown in Fig. 5 is considered. Therefore, we have (see detailed proof in Appendix A)

$$F^{SnF}_{(s,d)}(N, N_s, L_s) = \begin{cases} \prod_{l=1}^{L_s} [1-(1-p_s)^{L_s-l}(1-p_b) \\ \quad (1-F^{SnF}_{(i_1,d)}(N-1, N_s-1, l))] & \text{if } N_s > 1 \\ F^{AR}_{(s,d)}(N, L_s) & \text{if } N_s = 1 \end{cases} \tag{6}$$

B. EVALUATION OF PERFORMANCE AND COMPLEXITY

Here, we investigate the properties of $P^{SnF}_{(s,d)}(N, N_s, L_s)$ and $F^{SnF}_{(s,d)}(N, N_s, L_s)$, and compare the partial SnF scheme with the full SnF scheme. To gain a better understanding of the impact of N_s, two metrics are introduced. Performance ratio (PR) is defined as the ratio of $F^{SnF}_{(s,d)}(N, L)$ over $F^{SnF}_{(s,d)}(N, N_s, L_s)$, i.e., Eq. (7). Complexity ratio (CR) is defined as the ratio of $P^{SnF}_{(s,d)}(N, N_s, L_s)$ over $P^{SnF}_{(s,d)}(N, L)$, i.e., Eq. (8).

$$PR = \frac{F^{SnF}_{(s,d)}(N, L)}{F^{SnF}_{(s,d)}(N, N_s, L_s)} \tag{7}$$

$$CR = \frac{P^{SnF}_{(s,d)}(N, N_s, L_s)}{P^{SnF}_{(s,d)}(N, L)} \tag{8}$$

Herein, we first investigate the case when L_s=L. Figs. 6(a)-(c) show the numerical results given N=6 and L_s=L=4. In Fig. 6(a), $P^{SnF}_{(s,d)}(N, N_s, L_s)$ increases with N_s. The more storage nodes, the more alternate paths offered by the partial scheme. In Fig. 6(b), $F^{SnF}_{(s,d)}(N, N_s, L_s)$ decreases with N_s. Given p_b=0.01, $F^{SnF}_{(s,d)}(N, N_s, L_s)$ lies very close to $F^{SnF}_{(s,d)}(N, L)$ even when N_s=2. Fig. 6(c) shows that both PR and CR increase with N_s. The increase in PR becomes more evident when p_b decreases. By contrast, the increase in CR is independent of p_b and p_s. This suggests that the value of N_s required to achieve desirable performance decreases with p_b decreasing. For example, when p_b=0.3, at least four storage nodes (N_s=4) are required to achieve a PR of over 0.9. However, when p_b decreases to 0.01, only two storage nodes (N_s=2) can obtain a PR of over 0.9. Meanwhile, CR decreases from 0.625 to 0.179 when N_s decreases from 4 to 2. In other words, compared to the full scheme, over 0.9 of the original performance can be obtained at the cost of only 0.179 of the original complexity, when p_b=0.01. This implies that when the residual bandwidth is sufficient, desirable performance can be attained by only considering a few storage nodes.

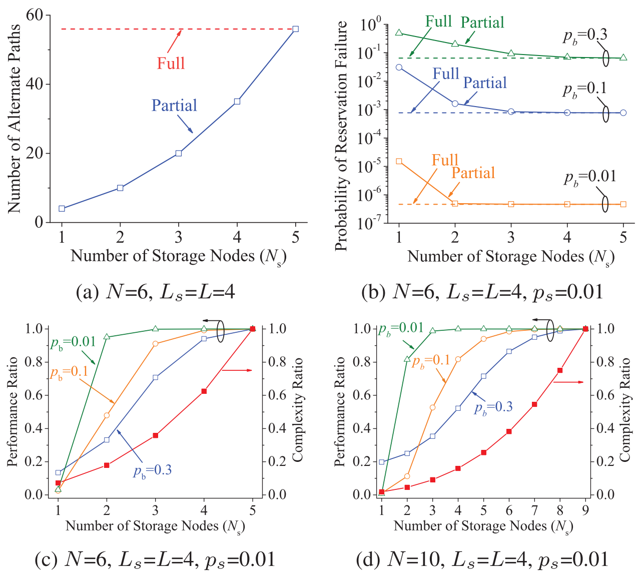

FIGURE 6. (a) Number of alternate paths. (b) Probability of reservation failure. Performance ratio and complexity ratio (c) and (d).

This can greatly reduce the computational burden. Similar results are obtained when N=10 and L=4 in Fig. 6(d).

Although both PR and CR increase with N_s, there exists a gap between PR and CR. This gap varies with N_s. Thus, the optimum number of storage nodes should lie in the region where PR is maximized and CR is minimized. However, this gap is bounded, since $1 \le N_s \le N-1$. This inspires us to further widen this gap by increasing L_s instead of N_s.

TABLE 2. Complexity Ratio and Performance Ratio (N=10, L=4 and p_s=0.01).

	$N_s=2$			$N_s=4$		
L_s	CR	PR [a]	PR [b]	CR	PR [a]	PR [b]
4	0.045	0.112	0.250	0.159	0.818	0.522
5	0.068	0.652	0.302	0.318	6.755	1.196
6	0.095	4.243	0.378	0.573	51.998	3.250
7	0.127	29.12	0.488	0.955	374.488	9.619

[a] p_b=0.1
[b] p_b=0.3

In the previous studies, L_s is equal to L. Here, we investigate the case when $L_s > L$. Consider N=10, L=4 and p_s=0.01. We investigate how L_s affects PR and CR. Numerical results are shown in Table 2. Given L_s=L=4, when N_s increases from 2 to 4, CR increases from 0.045 to 0.159. Meanwhile, PR increases from 0.112 to 0.818 when p_b=0.1, and PR increases from 0.25 to 0.522 when p_b=0.3, as shown in Row 3 of Table 2. As we can see, increasing N_s can increase PR up to 0.818 but at the cost of CR of 0.159 when p_b=0.1. By contrast, given L=4 and N_s remains 2, when L_s increases from 4 to 7, CR increases from 0.045 to 0.127. Meanwhile, PR increases from 0.112 to 29.12 when p_b=0.1, and PR increases from 0.25 to 0.488 when p_b=0.3, as shown in Row 6 of Table 2. Compared to increasing N_s, increasing L_s can increase *PR* while maintaining a lower *CR*.

C. LESSONS LEARNED FROM ANALYSIS

In short, our key findings are as follows:

1) Involving a portion of nodes rather than all nodes along the routing path in scheduling can reduce the computational burden, but at the cost of the degraded performance. In this case, given the same traffic load, the storage usage on each storage node may increase, which, in turn, increases the value of p_s.
2) When $p_s \ll p_b$, expanding the time horizon of the temporal scheduling is more useful than involving more nodes in scheduling for the performance improvement. However, with a larger time horizon, requests may experience longer delay, which will be elaborated upon in Sect. VI-A.
3) When $N_s < N$ and $L_s > L$, the partial scheme has the potential to outperform the full scheme while maintaining lower complexity. In other words, high performance as well as low complexity can be simultaneously achieved by involving a portion of nodes rather than all nodes along the routing path in scheduling.
4) The performance-complexity tradeoff in SnF should be jointly optimized by both N_s and L_s. But it comes at the cost of longer delay and higher storage usage.

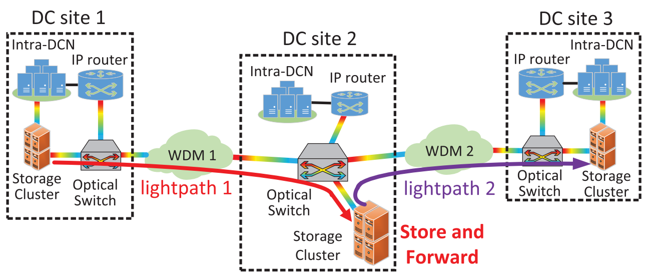

FIGURE 7. SnF scheduling bulk data over an inter-DCN.

V. PARTIAL SnF SCHEDULING METHOD

A. NETWORK MODEL AND ASSUMPTIONS

A Wavelength-Division Multiplexing (WDM) network is considered as the infrastructure layer for the inter-DC optical network. The network operator can coordinate the optical infrastructure and the DC resources to support the inter-DC bulk transfers by using transport-aware optimization with software-defined networking. Bandwidth resources on the optical fibers are allocated based on wavelength channels. The optical switch in each DC site is capable of OEO and wavelength conversions. The links are bidirectional. We assume that DC sites can store bulk data at their storage clusters. The storage cluster is designed to offer high-speed network paths for bulk transfers by bypassing enterprise firewalls [28], [29]. In Fig. 7, bulk data from DC site 1 are transferred over WDM 1 and stored on the storage cluster of DC site 2. Then, the bulk data are transferred to DC site 3 over WDM 2 when WDM 2 is less congested. The propagation delay between two sites is negligible when compared with transmission delays. A request is blocked when, upon arrival, no path exists between the source and the destination. Each request occupies one wavelength for its transfer. The processing overhead (e.g., the time required to perform network reconfiguration) is assumed to be negligible when compared with the transmission delay [4], [30]. Because bulk data flows will be served with dedicated OCS resources [31], we assume that a portion of the overall network resources are dedicated to the optical circuits that carry bulk data flows. The disk read/write speed is assumed to be equal to the transmission capacity of one wavelength. The infrastructure of DCN may face great challenges of link over-subscription, loops, TCP incast and outcast problems, especially when the disk read/write rates are high. Fortunately, novel transmission technologies and traffic control methods summarized in [32] may be used to overcome these challenges. For example, a TCAM (Ternary Content Addressable Memory) based routing technique was proposed to augment the routing capabilities of large-scale DCNs [33]. To tackle the outcast issue, a receiver-driven fair congestion control was proposed in [34]. A dynamic fair-share buffer policy for DC storage clusters was proposed to tackle the incast issue [35].

B. OVERVIEW

Inspired by the findings in Sect. IV, we incorporate the concepts of the partial SnF scheme and the network abstraction into routing with the TS-MLGs to realize a combination of $N_s < N$ and $L_s > L$, and hence propose a partial SnF (pSnF) scheduling method.

The main ideas of the pSnF method are twofold. First, upon the arrival of a request, the pSnF method involves a portion of nodes along the routing path in the scheduling decision making. Second, the pSnF method provides an abstract view of the network resources by merging spatial links connecting the involved nodes at each layer and condensing the redundant layers in the TS-MLG. On the one hand, the use of abstraction further reduces the size of the TS-MLG, and hence reduces the computational burden on scheduling. On the other hand, when considering a certain limit of the number of layers used for routing, the pSnF method can search the layers further ahead in time than the conventional method without the abstraction. This suggests that given the same limit of the computational complexity, requests with the pSnF method can reserve bandwidth and storage resources further ahead in time, i.e., essentially obtain a large time horizon of the temporal scheduling, when compared to requests with the conventional method. As a result, the pSnF method obtains high performance while simultaneously maintaining low complexity.

The pSnF method consists of five steps, as shown in Algorithm 1. K shortest routes for each node pair (i, j) have been pre-computed and stored in $\mathbb{R}$. Consider a request r for a source-destination pair (s, d). First, line 4 selects a pre-computed route $\boldsymbol{R_k}$ for (s, d) from set $\mathbb{R}$, where $\boldsymbol{R_k} \in \mathbb{R}$ and

Algorithm 1 Partial SnF (pSnF) Scheduling Method

```
1:  Input: $r = \{s, d\}, G, K, \mathbb{R}, \alpha$
2:  Output: $Path$, $Find$
3:  for $k = 1; k \leq K; k++$ do
4:      Select a route $\boldsymbol{R_k}$ from $s$ to $d$, where $\boldsymbol{R_k} \in \mathbb{R}$ and $|\boldsymbol{R_k}| = N_k$
5:      Select $N_s$ nodes among $\boldsymbol{R_k}$ based on a storage selection scheme, and get a segment set $\boldsymbol{S} = [s, I_1, \ldots, I_i, \ldots, I_{N_s-1}, d]$, where $N_s = \lceil (N_k - 1) \times \alpha \rceil$, $I_i \in \boldsymbol{R_k}$ and $|\boldsymbol{S}| = N_s + 1$
6:      Create a reduced subgraph $G'$ of $G$ based on $\boldsymbol{R_k}$
7:      Execute Algorithm 2 to get an abstract view $G''$ of $G'$ based on $\boldsymbol{S}$
8:      Execute BFS algorithm on $G''$ to find a viable path $Path$
9:      if $Path$ is valid then
10:         return $Path$ and $Find$ = True
11:     end if
12: end for
13: No viable path is found and return $Find$ = False
```

Algorithm 2 Storage-Based Network Abstraction Algorithm

```
1:  Input: $G'$, $L_R$, $\boldsymbol{S}$
2:  Output: $G''$
3:  Create an auxiliary graph $G'' = (V'', E'')$, where $V'' = \boldsymbol{S}$ and $E'' = \varnothing$
4:  for all the layers $\mathbb{L} = [l_1, l_2, \ldots, l_j, \ldots, l_{L_R}]$ in $G'$ do
5:      for each segment $[I_i, I_{i+1}]$ at layer $l_i$ do
6:          Find the spatial link $e_i$ with minimum residual bandwidth from node $I_i$ to node $I_{i+1}$ at layer $l_i$ in $G'$
7:          Add $e_i$ to link $<I_i, I_{i+1}>$ at layer $l_i$ in $G''$
8:          Add the temporal links connecting node $I_i$ and node $I_{i+1}$ in $G'$ to $G''$
9:      end for
10: end for
11: for all the layers $\mathbb{L} = [l_1, l_2, \ldots, l_j, \ldots, l_{L_R}]$ in $G''$ do
12:     if $l_{j+1} == l_j$ then
13:         Remove layer $l_{j+1}$ from $G''$
14:     end if
15: end for
16: return $G''$
```

$|\boldsymbol{R_k}| = N_k$. Second, line 5 selects N_s nodes along $\boldsymbol{R_k}$ for SnF scheduling based on a storage selection scheme. Let α denote the percentage of the nodes on $\boldsymbol{R_k}$ used for scheduling, where $0 < \alpha \leq 1$. We have $N_s = \lceil (N_k - 1) \times \alpha \rceil$. Set $\boldsymbol{S}$ denotes the set of segments when $\boldsymbol{R_k}$ is divided by the storage nodes into N_s segments. $\boldsymbol{S} = [s, I_1, \ldots, I_i, \ldots, I_{N_s-1}, d]$, where I_i denotes a storage node, $I_i \in \boldsymbol{R_k}$ and $|\boldsymbol{S}| = N_s + 1$. The first involved node is s, because prior studies showed that majority of SnF operations are performed on the sources [14], [22], [23]. For simplicity, the other involved nodes are considered to be equally spaced along $\boldsymbol{R_k}$. Exploring a network-level storage selection scheme is an interesting, yet complicated, problem that is worthy of further study. Third, line 6 reduces the original TS-MLG G to a smaller graph G', based on $\boldsymbol{R_k}$. Specifically, G' only contains nodes in $\boldsymbol{R_k}$ and links that connect these nodes. Fourth, line 7 executes Algorithm 2 to get an abstract view G'' of G' based on $\boldsymbol{S}$. G'' denotes a condensed subgraph of G'. Fifth, line 8 executes the Breadth-First Search (BFS) algorithm on G'' to find a viable path (i.e., *Path*). If the BFS algorithm fails to find any viable path on $\boldsymbol{R_k}$, the pSnF method re-runs with the next route $\boldsymbol{R_{k+1}}$. If no path is found among the K pre-computed routes, r is blocked. Various routing algorithms (e.g., Dijsktra's algorithm) can be applied on G'' to search a viable path. In this paper, BFS is used to find the first viable path reaching the destination d, because requests are assumed to prefer to be transferred as soon as possible.

The main idea of Algorithm 2 is to provide an abstract view of G' based on $\boldsymbol{S}$, which consists of three steps. First, line 3 creates an auxiliary graph $G'' = (V'', E'')$, where $V'' = \boldsymbol{S}$ and $E'' = \varnothing$. Second, lines 4-8 merges the network states of spatial links within each segment in G' into a logical link in G''. L_R denotes the number of layers used for routing. $\mathbb{L}$ denotes the set of these layers, where $\mathbb{L} = [l_1, \ldots, l_j, \ldots, l_{L_R}]$. For each segment $[I_i, I_{i+1}]$ at layer l_j in G', lines 5-8 find a spatial link e_i with the minimum residual bandwidth within $[I_i, I_{i+1}]$, add e_i to link $<I_i, I_{i+1}>$ at layer l_j in G'', and add the temporal links connecting node I_i and node I_{i+1} in G' to G''. Third, lines 10-11 condense the redundant layers in G'. If adjacent layers represent the same network state, they are condensed on a single layer.

The flow chart of the provisioning process with the pSnF method is depicted in Fig. 8. Upon the arrival of request r, the expired layers in G will be removed. Then, the pSnF method is executed to find a viable path on G. If a viable path is found, the resources along *Path* will be reserved by updating the existing layers and adding new layers to G. Otherwise, request r will be rejected.

C. EXAMPLE

Here, we demonstrate how the pSnF method reduces the size of the TS-MLG through an example depicted in Figs. 9 and 10. The original TS-MLG G is shown in Fig. 2. G consists of multiple layers, with each layer containing six vertices. Consider a request r from node 1 to node 4. A pre-computed shortest route, say, $\boldsymbol{R_k}$=$\{1, 2, 3, 4\}$, is selected for r. First, G is reduced to G' based on $\boldsymbol{R_k}$. Fig. 9(a) shows the resulting graph G', which consists of four layers, with each layer containing four vertices. In Fig. 9(a), the first three nodes on $\boldsymbol{R_k}$ are connected via temporal links, when all the storage nodes are considered. Assume α=0.4 and hence get N_s=2. Based on the aforementioned selection scheme, node 1 and node 3 are involved in scheduling, i.e., $\boldsymbol{S} = [1, 3, 4]$. To highlight this, the temporal links connecting node 2 are omitted, as shown in Fig. 9(b). This will not be actually executed in Algorithm 1.

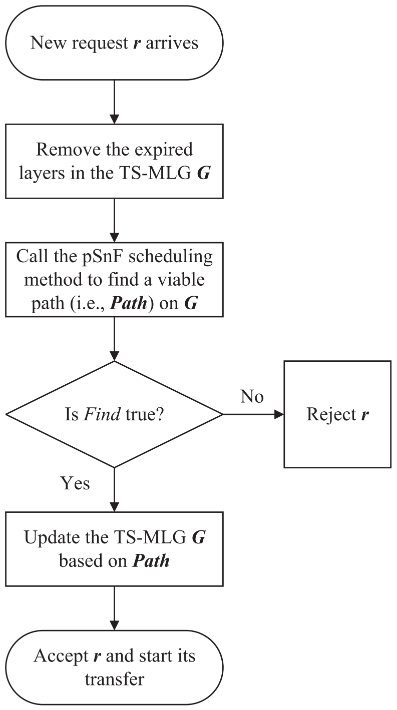
New request *r* arrives
Remove the expired layers in the TS-MLG *G*
Call the pSnF scheduling method to find a viable path (i.e., *Path*) on *G*
Is *Find* true?
No
Reject *r*
Yes
Update the TS-MLG *G* based on *Path*
Accept *r* and start its transfer

FIGURE 8. The flow chart of the provisioning process.

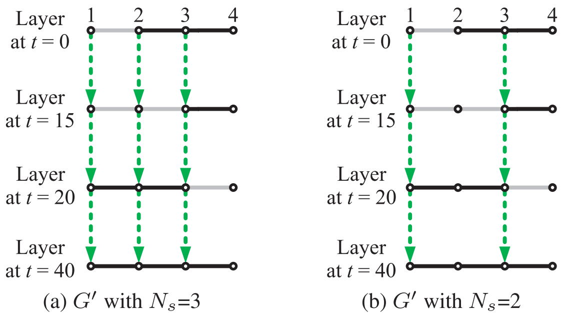

FIGURE 9. The reduced TS-MLG G'. (a) G' with N_S=3 and (b) G' with N_S=2.

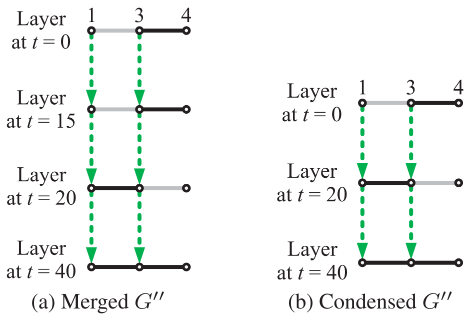

FIGURE 10. (a) The merged TS-MLG G'' with 4 layers and 3 vertices in each layer. (b) The condensed TS-MLG G'' with 3 layers and 3 vertices in each layer.

Then, Algorithm 2 is executed to merge the spatial links within the segment [1, 3]. Fig. 10(a) shows the resulting TS-MLG G'', which consists of four layers, with each layer containing three vertices. The spatial link between node 1 and node 3 at each layer in Fig. 10(a) represents the minimum residual bandwidth from node 1 to node 3 at different instants. For example, in Fig. 9(b), link 1-2 is busy at t=0, while link 2-3 is free at t=0. Thus, in Fig. 10(a), link 1-3 is busy at t=0, since the minimum residual bandwidth within [1, 3] is zero at t=0.

Moreover, in Fig. 10(a), the layers at t=0 and t=15 in G' represent the same network state. These redundant layers cannot provide more useful information, but impose extra computational burden on searching. Thus, Algorithm 2 removes the redundant layer at t=15. The resulting graph G'' is shown in Figs. 10(b). The resulting G'' consists of three layers, with each layer containing three vertices. By performing the pSnF method, the size of the TS-MLG is greatly reduced.

D. RESERVATION WINDOW

As mentioned in Sect. II-B, to bound the computational complexity, a request can only search the path within a given number of layers (i.e., L_R). The value of L_R implies the horizon of temporal scheduling. Thus, when the links and the layers are merged and condensed by Algorithm 2, the horizon of temporal scheduling changes correspondingly. To study this, the reservation window is defined as the time interval between the topmost layer and the L_R-th layer (i.e., the last layer that can be used for routing). A larger window size increases the chances of finding available resources for requests. The window size is related to both L_R and the time interval between the layers.

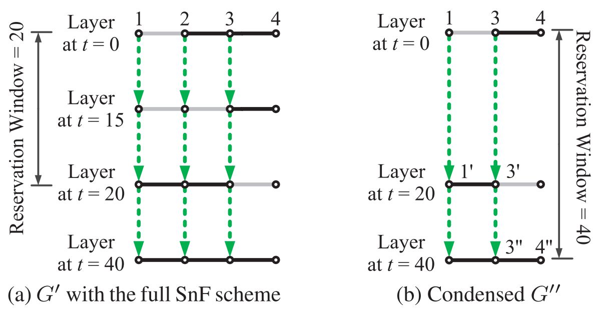

FIGURE 11. Reservation windows in the G' with the full SnF scheme (a) and the condensed G'' when L_R=3 (b).

We compare the G' shown in Fig. 9(a) with the condensed G'' shown in Fig. 10(b) in terms of the reservation window. Assume L_R is equal to 3. Fig. 11 shows the comparison. In Fig. 11(a), the reservation window in G' is the time interval between the topmost layer and the third layer, i.e., the layer at t=0 and the layer at t=20. The window size in Fig. 11(a) hence is 20. In Fig. 11(b), the reservation window in G'' is also the time interval between the topmost layer and the third layer. However, because the layer at t=15 is removed, the third layer is the layer at t=40 instead of that at t=20. The window size in Fig. 11(b) hence is 40. Consider a request r from node 1 to node 3 arrives at the network at t=0. No viable path can be found within the window shown in Fig. 11(a). Thus, r is blocked. By contrast, a path 1-1'-3'-3''-4'' is available to deliver r within the window shown in Fig. 11(b).

In short, given the same limit of the computational complexity, compared to the conventional method, the pSnF method provides a wider reservation window, which can benefit the blocking probability.

E. COMPUTATIONAL COMPLEXITY

The complexity of routing with the TS-MLG is dominantly determined by the TS-MLG size, which is $O((V \cdot L_R)^2)$ [14]. V denotes the total number of vertices in the network topology. In the pSnF method, the BFS algorithm is executed to find a viable path in the condensed TS-MLG G''. The complexity of the BFS algorithm is $O(V'' + E'')$. V'' denotes the total number of vertices in G'' and E'' denotes the total number of edges in G''. In the worst case, V'' is equal to $L_R \cdot N_s$ and E'' is equal to the sum of both temporal links, i.e., $(L_R - 1) \cdot N_s$, and spatial links, i.e., $(N_s - 1) \cdot L_R$. Thus, the complexity of the pSnF method is $O(K \cdot L_R \cdot N_s)$.

VI. EVALUATIONS AND DISCUSSIONS

In this section, dynamic network environments are simulated to compare the pSnF method with the conventional SnF scheduling method using the full SnF scheme [12] (i.e., the full method) under various network scenarios.

We apply the major assumptions described in Sect. V-A, and use the NSFNET topology in our study. For simplicity, we relax the storage capacity constraints. Thus, storage capacity is not a concern in routing requests. Requests are accepted or blocked, depending on whether viable paths are found in the TS-MLG within a given number of layers, i.e., L_R. Request arrivals are assumed to be independent and uniformly distributed among all source-destination pairs; the arrival process is Poisson with rate λ. In other words, λ requests arrive at the network per unit time. Request durations are assumed to be exponentially distributed with rate μ. In other words, μ requests depart the network per unit time. Traffic load $\rho = \lambda/\mu$. Spatial and temporal links are assigned with the same cost, i.e., 1 unit. The routing problem herein is therefore formulated as a shortest-hop-count problem. The link capacity, i.e., the number of wavelengths per link, is denoted as w, and K denotes the number of shortest alternate routes for each node pair. The percentage of nodes along a route involved in scheduling is denoted as α. When α=1, all nodes along a route are involved in scheduling. In this case, the pSnF method is equivalent to the full method. For simplicity, α is set to 0.4 and 0.6. We average the results over 20 simulation runs, each with 500,000 requests.

A. TRAFFIC LOAD

We first investigate how the blocking probability varies with ρ, which can be increased by either increasing λ or by decreasing μ. The results obtained in both cases are similar. Thus, in the following simulations, $\lambda = 1$, and we increase ρ by varying μ.

Fig. 12 shows the simulation results. The blocking probability increases when ρ increases from 10 to 60. In Fig. 12(a), given w=4, L_R=4 and K=3, when ρ=10, the pSnF method

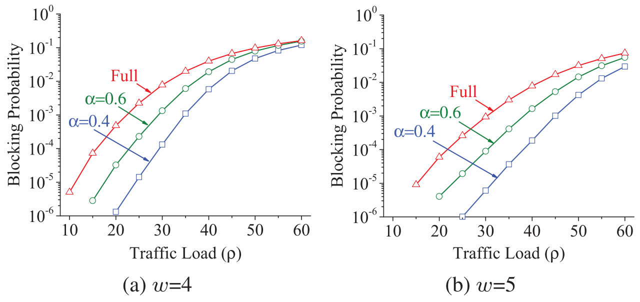

FIGURE 12. Blocking probability under various values of ρ in NSFNET (L_R=4, K=3).

with α=0.4 and 0.6 yields a blocking probability of 0. By contrast, the full method yields a blocking probability of 5.05×10^{-6}. With ρ increasing, request blocking begins to occur in the cases of the pSnF method with $\alpha = 0.4$ and 0.6. The larger value of α, the more evident increase in the blocking probability. The results in Fig. 12(b) follow similar trends, but request blocking occurs at larger values of ρ, because w is higher in Fig. 12(b) than in Fig. 12(a).

Intuitively, the fewer storage nodes involved, the higher blocking probability. However, the results in Fig. 12 are opposite. The fewer storage nodes involved, the lower blocking probability the pSnF method can obtain. To understand this finding, we focus on the result in Fig. 12(a), and investigate how various performance metrics change with ρ. The results are depicted in Fig. 13. Active request is defined as a request which is accepted but not yet finished. Link utilization is defined as the ratio of the used bandwidth over the total bandwidth for all spatial links during the simulation span. Ratio of stored requests is defined as the ratio between the number of the stored requests and the total number of requests. Delay is defined as the time interval between the request arrival instant and the transfer completion instant. The average delay shown in Fig. 13(e) is averaged over all successful requests.

In Fig. 13(a), both the pSnF method and the full method obtain similar link utilization when ρ increases from 10 up to 30. When ρ increases beyond 35, the pSnF method with α=0.6 yields higher link utilization than the others; while the full method yields lower link utilization than the others. By contrast, the increase in the number of active flows is greater in the pSnF method with α=0.4 when compared to the others for ρ larger than 30, as seen in Fig. 13(b). This suggests that when the traffic load is medium or higher, the network can accommodate more requests while maintaining low utilization in the pSnF method with α=0.4 than in the others. The smaller value of α, the more requests are accommodated in the network. This happens because the pSnF method with α=0.4 provides wider reservation windows than the others, as shown in Fig. 13(c). The smaller value of α, the fewer storage nodes involved, the more spatial links would be merged. This increases the chance of finding redundant layers in the TS-MLG. With more redundant layers being condensed, the pSnF method can search the layers further

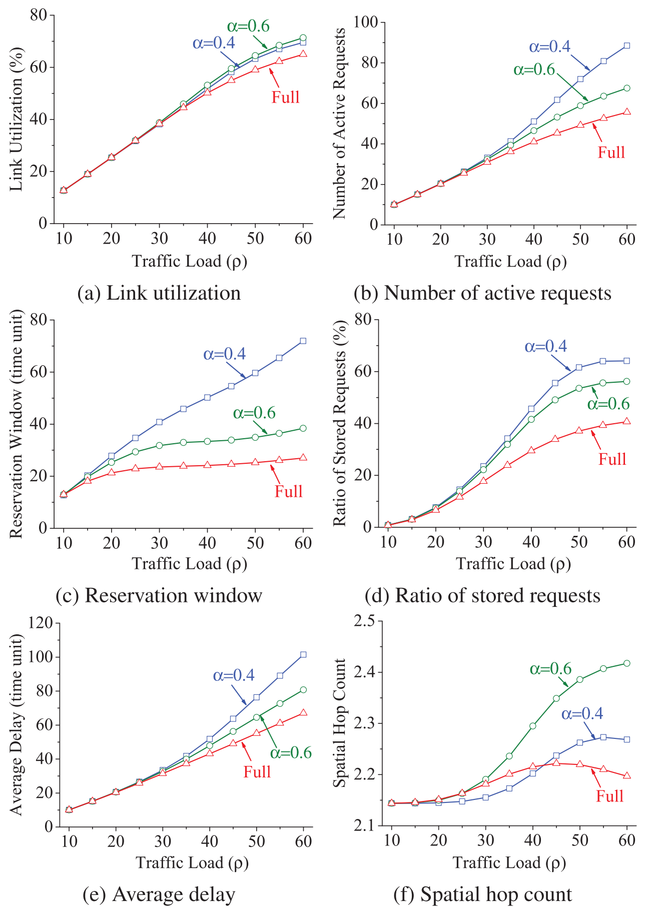

FIGURE 13. **Network performance metrics under various values of ρ (w=4, L_R=4 and K=3).**

ahead in time, given a certain value of L_R. Thus, the smaller value of α, the wider reservation window. As a result, more requests can be served through SnF, as shown in Fig. 13(d). In short, benefiting from the abstraction, requests are more likely to be served through SnF in the pSnF method than in the full method. However, this comes at the cost of longer delay, as shown in Fig. 13(e).

With the window size widening, more requests could be routed over short routes through SnF rather than detour over long routes. Fig. 13(f) shows that the average spatial hop counts of the accepted requests. The pSnF method with α=0.4 yields fewer spatial hops than that with α=0.6. The full method yields fewer spatial hops than the pSnF method when ρ=45∼60. This is because the full method suffers from high blocking probability than the pSnF method. Requests that need long-spatial-hop routes are more difficult to be satisfied in the full method. With more long-spatial-hop requests being blocked, the spatial hop count in the full method remains lower than the pSnF method.

B. IMPACT OF THE NETWORK ABSTRACTION

Benefiting from the network abstraction (i.e., Algorithm 2), spatial links and redundant layers are merged and condensed,

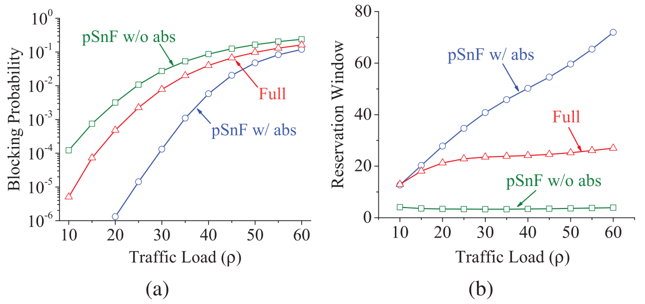

FIGURE 14. **Comparison between the original pSnF method and the pSnF method without the network abstraction (α=0.4).**

which enables a wider reservation window for SnF scheduling. To verify the impact of the network abstraction, we conduct a comparison between the original pSnF method and the pSnF method without the network abstraction (i.e., the modified pSnF method). In the modified pSnF method, Algorithm 2 is disabled and hence the BFS algorithm is executed to search viable paths on the reduced TS-MLG G' instead of the condensed TS-MLG G''. Herein, w=4, L_R=4 and K=3. Simulation results are shown in Fig. 14.

Given α=0.4, the original pSnF method outperforms the modified pSnF method in terms of blocking probability, as seen in Fig. 14(a). This occurs because the reservation window in the original pSnF method increases with ρ, while that in the modified pSnF method almost remains constant, as shown in Fig. 14(b). Then, we compare the modified pSnF method with the full method. As we can see, the full method outperforms the modified pSnF method in terms of the blocking probability, as shown in Fig. 14(a). This is because on the one hand, without the network abstraction, the modified pSnF method suffer a narrower reservation window than the full method, as seen in Fig. 14(b). On the other hand, the full method enables more storage nodes than the modified pSnF method.

C. COMPUTING TIME

We investigate the impact of the number of nodes and layers in the TS-MLG on the computing time of the pSnF method with different values of α. We use randomly generated topologies with density 0.6 for simulations. Here, the density is defined as the probability of an edge between any two nodes. V denotes the total number of nodes in the network topology.

Figs. 15(a) and (b) show the average computing time for different methods to perform the route search on a given TS-MLG. In Fig. 15(a), V is fixed to 10. The computing time increases with L_R. In Fig. 15(b), L_R is fixed to 10. The computing time increases with V. Fig. 15 shows that the smaller value of α, the less evident increase in the computing time. This happens because a smaller value of α enables a smaller size of the TS-MLG needed to be searched.

D. OTHER FACTORS

We first investigate the impact of the number of alternate routes, i.e., K, on the blocking probability. The simulation

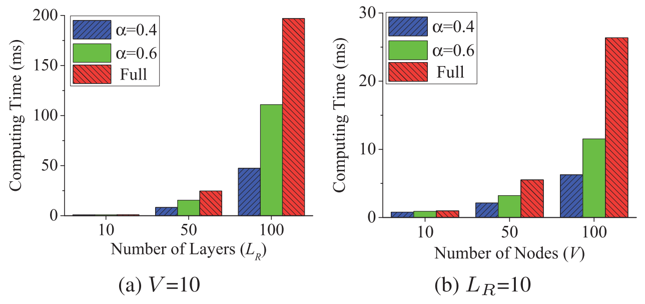

FIGURE 15. Computing time under various values of L_R when V=10 (a) and under various values of V when L_R=10 (b).

results are shown in Fig. 16(a). Herein, w=4 and L_R=4. Fig. 16(a) shows that given various K, the blocking probability increases with ρ. The larger value of K, the lower blocking probability. In Fig. 16(b), we investigate the impact of the link capacity (w). Herein, K=3, L_R=4 and ρ=40. Fig. 16(b) shows that the blocking probability decreases with w. The smaller value of α, the more evident decrease.

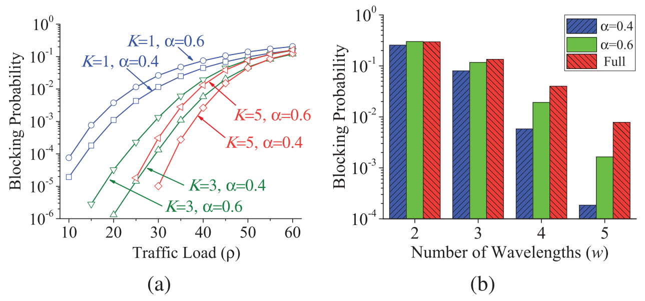

FIGURE 16. Blocking probability given various values of K (a) and w (b).

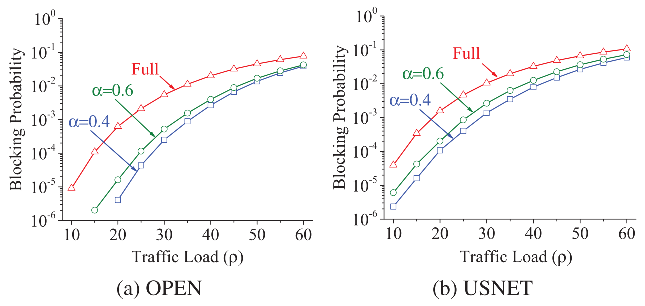

FIGURE 17. Blocking probability in OPEN (a) and in USNET (b).

Then, we investigate how the blocking probability varies with ρ in different network topologies. The 19-node 39-link optical pan-european network (OPEN) and the 24-node 43-link US backbone network (USNET) are used. Herein, w=4, L_R=4 and K=3. Figs. 17(a) and (b) depict the blocking probability under various ρ in OPEN and USNET, respectively. The results in Fig. 12 and Fig. 17 follow similar trends. Therefore, the pSnF method can outperform the full method in different topologies. However, the blocking probability in Fig. 17 is slightly higher than that in Fig. 12. This is because OPEN and USNET are larger than NSFNET. The average spatial hop count hence are larger in OPEN and USNET than that in NSFNET. Given the same value of α, more storage nodes are selected along each route in OPEN and USNET than in NSFNET. Redundant layers are more likely to be found and condensed when fewer storage nodes along each route are selected. Thus, with the network abstraction, requests can obtain a wider reservation window in NSFNET than in OPEN and USNET. As a result, given the same value of α, a large network topology may have a narrower reservation window and hence expect more request blocking, compared to a small network topology.

VII. CONCLUSION

In this paper, we compare SnF with IR and AR. Our studies reveal that choosing the number of storage nodes involved in scheduling is a tradeoff between performance and complexity. The analytic models are presented to quantify the impact of the number of storage nodes on this tradeoff. Our key findings show that the partial SnF scheme with a larger time horizon of the temporal scheduling has the potential to outperform the full SnF scheme while maintaining lower complexity.

Thus, we propose a pSnF scheduling method. The pSnF method involves a portion of nodes along the routing path in scheduling. Meanwhile, the pSnF method introduces a storage-based network abstraction. Given the same limit of the computational cost, the pSnF method can reserve resources further ahead in time than the conventional full method. Simulations demonstrate that the pSnF method has lower computing time while achieving better blocking performance when compared to the full method.

The analytic models in this work focus on the case of a fixed route and hence are not sophisticated enough to gain practical insights into the impact of network-level storage selection schemes. Extending the analytic models to the case of general networks would be an exciting research direction to explore in the future.

APPENDIX

PROOF OF EQUATION 6

First, let us consider the case when $N_s > 1$. In this case, the proof of Eq. (6) can be carried out with the following detailed steps:

- The TS-MLG of $F^{SnF}_{(s,d)}(N, N_s, L_s)$ could be decomposed into L_s different cases.
- $F^{SnF}_{(i_1,d)}(N-1, N_s-1, l)$ denotes the TS-MLG of the l-th case, where $l \in [1, L_s]$.
- In the l-th case, all the alternate paths traverse $L_s - l$ temporal links and one spatial link before reaching node i_1.
- Then, they diverge in the TS-MLG of $F^{SnF}_{(i_1,d)}(N-1, N_s-1, l)$, as shown in Fig. 18.
- Thus, we have the reservation failure probability of the l-th case, i.e., $1-(1-p_s)^{L_s-l}(1-p_b)\times(1-F^{SnF}_{(i_1,d)}(N-1, N_s-1, l))$.

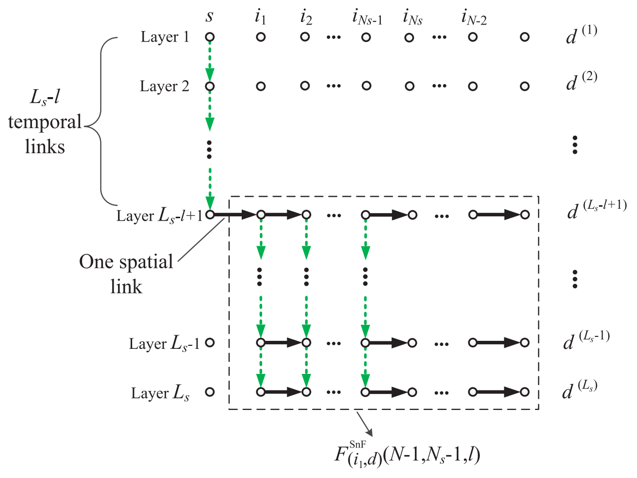

FIGURE 18. TS-MLG of $F^{SnF}_{(i_1,d)}(N-1, N_s-1, L_s)$, i.e., the i-case when decomposing $F^{SnF}_{(s,d)}(N, N_s, L_s)$ and $N_s > 1$.

- We assume that these L_s cases are independent. In other words, a request will fail only when it cannot find any viable path from all the L_s cases. Thus, we have a lower bound of $F^{SnF}_{(s,d)}(N, N_s, L_s)$.
- We get $F^{SnF}_{(s,d)}(N, N_s, L_s) = \prod_{l=1}^{L_s}[1 - (1 - p_s)^{L_s-l} (1 - p_b) \times (1 - F^{SnF}_{(i_1,d)}(N - 1, N_s - 1, l))]$.

Then, let us consider the case when $N_s = 1$. In this case, the proof of Eq. (6) can be carried out with the following detailed steps:

- We assume that the first node is selected as the storage node.
- Therefore, the TS-MLG of $F^{SnF}_{(s,d)}(N, N_s, L_s)$ is equivalent to the TS-MLG of in Fig. 3(b).
- We get $F^{SnF}_{(s,d)}(N, N_s, L_s) = F^{AR}_{(s,d)}(N, L_s)$, where the expression of $F^{AR}_{(s,d)}(N, L_s)$ has been given in Eq. (2).

REFERENCES

[1] X. Lu, F. Kong, X. Liu, J. Yin, Q. Xiang, and H. Yu, "Bulk savings for bulk transfers: Minimizing the energy-cost for geo-distributed data centers," *IEEE Trans. Cloud Comput.*, vol. 8, no. 1, pp. 73–85, Jan. 2020.

[2] J. L. Garcia-Dorado and S. G. Rao, "Cost-aware multi data-center bulk transfers in the cloud from a customer-side perspective," *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 34–47, Jan. 2019.

[3] L. Luo, Y. Kong, M. Noormohammadpour, Z. Ye, G. Sun, H. Yu, and B. Li, "Deadline-aware fast One-to-Many bulk transfers over inter-datacenter networks," *IEEE Trans. Cloud Comput.*, early access, Aug. 20, 2019, doi: .

[4] Z. Yang, Y. Cui, X. Wang, Y. Liu, M. Li, S. Xiao, and C. Li, "Cost-efficient scheduling of bulk transfers in inter-datacenter WANs," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1973–1986, Oct. 2019.

[5] W. Li, X. Zhou, K. Li, H. Qi, and D. Guo, "TrafficShaper: Shaping inter-datacenter traffic to reduce the transmission cost," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1193–1206, Jun. 2018.

[6] M. Noormohammadpour, C. S. Raghavendra, S. Kandula, and S. Rao, "QuickCast: Fast and efficient inter-datacenter transfers using forwarding tree cohorts," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 225–233.

[7] W. Hu, J. Liu, T. Huang, and Y. Liu, "A completion time-based flow scheduling for inter-data center traffic optimization," *IEEE Access*, vol. 6, pp. 26181–26193, 2018.

[8] Y. Wu, Z. Zhang, C. Wu, C. Guo, Z. Li, and F. C. M. Lau, "Orchestrating bulk data transfers across geo-distributed datacenters," *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 112–125, Jan. 2017.

[9] P. Lu and Z. Zhu, "Data-oriented task scheduling in Fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightw. Technol.*, vol. 35, no. 24, pp. 5335–5346, Dec. 15, 2017.

[10] C. Sun, W. Guo, Z. Liu, M. Xia, and W. Hu, "Performance analysis of storage-based routing for circuit-switched networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 8, no. 5, pp. 282–289, May 2016.

[11] Y. Wang, S. Su, A. X. Liu, and Z. Zhang, "Multiple bulk data transfers scheduling among datacenters," *Comput. Netw.*, vol. 68, pp. 123–137, Aug. 2014.

[12] X. Lin, X. Wang, S. Yue, W. Sun, M. Veeraraghavan, and W. Hu, "Design of an SNF scheduling method for bulk data transfers over inter-datacenter WANs," in *Proc. IEEE 20th Int. Conf. High Perform. Switching Routing (HPSR)*, May 2019, pp. 1–8.

[13] X. Lin, W. Sun, X. Wang, S. Yue, M. Veeraraghavan, and W. Hu, "Time-space decoupled SnF scheduling of bulk transfers across inter-datacenter optical networks," *IEEE Access*, vol. 8, pp. 24829–24846, 2020.

[14] X. Lin, W. Sun, M. Veeraraghavan, and W. Hu, "Time-shifted multilayer graph: A routing framework for bulk data transfer in optical circuit-switched networks with assistive storage," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 8, no. 3, pp. 162–174, Mar. 2016. [Online]. Available: http://jocn.osa.org/abstract.cfm?URI=jocn-8-3-162

[15] A. Patel, M. Tacca, and J. P. Jue, "Time-shift circuit switching," in *Proc. Conf. Opt. Fiber Commun./Nat. Fiber Optic Eng. Conf. (OFC/NFOEC)*, Feb. 2008, p. OThI6.

[16] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 74–85, 2011.

[17] Y. Li, H. Wang, P. Zhang, J. Dong, and S. Cheng, "D4D: Inter-datacenter bulk transfers with ISP friendliness," in *Proc. IEEE Int. Conf. Cluster Comput.*, Sep. 2012, pp. 597–600.

[18] Y. Feng, B. Li, and B. Li, "Postcard: Minimizing costs on inter-datacenter traffic with Store-and-Forward," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2012, pp. 43–50.

[19] P. Chhabra, V. Erramilli, N. Laoutaris, R. Sundaram, and P. Rodriguez, "Algorithms for constrained bulk-transfer of delay-tolerant data," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.

[20] D. Feng, W. Sun, and W. Hu, "Joint provisioning of lightpaths and storage in store-and-transfer wavelength-division multiplexing networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. 218–233, Mar. 2017.

[21] D. Feng, W. Sun, X. Zhang, and W. Hu, "Dimensioning of the store-and-transfer WDM network with limited node storage under the sliding scheduled traffic model," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 4, pp. 275–290, Apr. 2017. [Online]. Available: http://jocn.osa.org/abstract.cfm?URI=jocn-9-4-275

[22] A. N. Patel, Y. Zhu, and J. P. Jue, "Routing and horizon scheduling for time-shift advance reservation," in *Proc. Opt. Fiber Commun. Conf. Nat. Fiber Optic Eng. Conf.*, 2009, p. OThO4. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=OFC-2009-OThO4

[23] A. N. Patel, Y. Zhu, Q. She, and J. P. Jue, "Routing and scheduling for time-shift advance reservation," in *Proc. 18th Int. Conf. Comput. Commun. Netw.*, Aug. 2009, pp. 1–6.

[24] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, and R. Sundaram, "Delay-tolerant bulk data transfers on the Internet," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1852–1865, Dec. 2013.

[25] C. Lee and J.-K. K. Rhee, "Efficient design and scalable control for store-and-forward capable optical transport networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 8, pp. 699–710, Aug. 2017.

[26] G. Iosifidis, I. Koutsopoulos, and G. Smaragdakis, "Distributed storage control algorithms for dynamic networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1359–1372, Jun. 2017.

[27] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1037–1064, Nov. 2012.

[28] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, "The science DMZ: A network design pattern for data-intensive science," in *Proc. IEEE/ACM Int. Conf. High Perform. Comput., Netw., Storage Anal. (SC)*, Nov. 2013, pp. 1–10.

[29] X. Wang, X. Lin, W. Sun, and M. Veeraraghavan, "Comparison of two sharing modes for a proposed optical enterprise-access SDN architecture," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Nov. 2018, pp. 1–8.
[30] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, no. 14, pp. 3005–3015, Jul. 15, 2015.
[31] C. Castillo, G. N. Rouskas, and K. Harfoush, "On the design of online scheduling algorithms for advance reservations and QoS in grids," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, 2007, pp. 1–10.
[32] W. Xia, P. Zhao, Y. Wen, and H. Xie, "A survey on data center networking (DCN): Infrastructure and operations," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 640–656, 1st Quart., 2017.
[33] M. K. Khattak, Y. Tang, H. Fahim, E. Rehman, and M. F. Majeed, "Effective routing technique: Augmenting data center switch fabric performance," *IEEE Access*, vol. 8, pp. 37372–37382, 2020.
[34] J. Huang, S. Li, R. Han, and J. Wang, "Receiver-driven fair congestion control for TCP outcast in data center networks," *J. Netw. Comput. Appl.*, vol. 131, pp. 75–88, Apr. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804519300311
[35] Y. A. Bangash, T. Rana, H. Abbas, M. A. Imran, and A. A. Khan, "Incast mitigation in a data center storage cluster through a dynamic fair-share buffer policy," *IEEE Access*, vol. 7, pp. 10718–10733, 2019.

XIAO LIN (Member, IEEE) received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University. He was a Visiting Scholar of the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia. He is currently an Assistant Professor with the College of Physics and Information Engineering, Fuzhou University (FZU). His research interests include intelligent edge computing, optical switches, reliability, and bulk data transfer over optical networks.

SHENGNAN YUE (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree in electronic information and electrical engineering with Shanghai Jiao Tong University (SJTU). Her research interests include bulk data transfer and delay-tolerant networks.

YUANLONG TAN received the master's degree from the State Key Laboratory of Information Photonics and Optical Communication, Beijing University of Posts and Telecommunications (BUPT), China. He is currently pursuing the Ph.D. degree with the Charles L. Brown Department of Electrical and Computer Engineering, University of Virginia (UVA). His research interests include network multicast, software-defined networks, elastic optical networks, and network survivability.

WEIQIANG SUN (Senior Member, IEEE) is currently a Full Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University (SJTU). He also works with the Shanghai Institute for Advanced Communication and Data Science, Shanghai, China. He is actively involved in the research of high-speed networks, network control and management, and network applications. He has about 60 publications in peer-reviewed journals and conferences. He was actively involved in the standardization of Generalized MultiProtocol Label Switching (GMPLS) performance measurement in IETF. He is a Co-Editor of RFC 6777 and RFC 5814. He has served as an Invited Speaker on many international conferences. He is a member of the Technical Program Committee of COIN, GreenTouch, PGC, and an Organizer of the Sino-Korea Workshop on IPTV and NGN, from 2007 to 2010. He is one of the TPC Co-Chairs of the IEEE HPSR 2017.

MALATHI VEERARAGHAVAN (Fellow, IEEE, In Memoriam) received her B. Tech degree from Indian Institute of Technology (Madras) in 1984, and MS and Ph.D. degrees from Duke University in 1985 and 1988, respectively. After a ten-year career at Bell Laboratories, she served on the faculty at Polytechnic University, Brooklyn, New York from 1999-2002, where she won the Jacobs award for excellence in education in 2002. She served as Director of the Computer Engineering Program at University of Virginia (UVA) 2003-2006. In 2007, she was promoted to a Professor with the Charles L. Brown Department of Electrical and Computer Engineering, UVA. In 2020, she was elevated to IEEE Fellow, "for contributions to control-plane architectures, signal protocols and hybrid networks." She holds 30 patents, has over 130 publications and has received seven Best paper awards. She served as the Technical Program Committee Co-Chair for the High-Speed Networking Symposium in IEEE ICC 2013, as Technical Program Committee Chair for IEEE ICC 2002 and Associate Editor for the IEEE/ACM Transactions on Networking. She was General Chair for IEEE Computer Communications Workshop in 2000, and served as an Area Editor for IEEE Communication Surveys. She served as Editor of IEEE ComSoc e-News and as an Associate Editor of the IEEE Transactions on Reliability from 1992–1994.

WEISHENG HU (Senior Member, IEEE) received the B.Sc. degree from Tsinghua University, in 1986, the M.Eng. degree from the University of Science and Technology Beijing, in 1989, and the Ph.D. degree from Nanjing University, in 1997. He has been a Postdoctoral Fellow with Shanghai Jiao Tong University, since 1997, and as a Professor, since 1999, where he was promoted to Distinguished Professor, in 2009. He has served as the Deputy Director and the Director of the State Key Laboratory of Advanced Optical Communication Systems and Networks, from 2002 to 2012. He has published about 400 peer-reviewed journal/conference papers. He serves on several editorial boards, including *Optics Express*, the *Journal of Lightwave Technology*, *Chinese Optics Letters*, and *China Communications*. He has served on the program committees of a number of international conferences, including OFC, ICC, and INFOCOM.