Adaptive Multi-Path SnF Scheduling Method for Delay-Sensitive Transfers Across Inter-Datacenter Optical Networks

Xiao Lin*, Shuo Ji*, Shengnan Yue†, Jun Li‡, Weiqiang Sun†, Weisheng Hu†
*College of Physics and Information Engineering,
Fuzhou University, Fuzhou, China, {linxiaocer,201127035}@fzu.edu.cn
†State Key Laboratory of Advanced Optical Communication Systems and Networks,
Shanghai Jiao Tong University, Shanghai, China, {shnyue,sunwq,wshu}@sjtu.edu.cn
‡School of Electronic and Information Engineering, Soochow University, Suzhou, China, ljun@suda.edu.cn

***Abstract*—Traffic fluctuations make conventional end-to-end delay-sensitive (DS) transfers difficult to fully utilize the residual bandwidth in the inter-datacenter networks. In this paper, we introduce datacenter storage and multi-path routing into DS transfers, and present a scheduling method. On one hand, it can split data into shorter segments and route them through multiple paths. On the other hand, it leverages temporary storage at intermediate sites to segment each path into sub-paths when provisioning the end-to-end lightpath fails. Studies show that compared with the state-of-the-art scheduling methods, the proposed method can accommodate more DS transfers, shorten delivery time and accelerate DS transfers efficiently.**

***Index Terms*—Multi-path routing, inter-datacenter networks, resource scheduling, store-and-forward, storage.**

I. Introduction

Bulk data, generated from real-time data analysis, traffic surveillance and disaster detecting system, are often delay-sensitive (DS) [1]. For example, the traffic surveillance information is sent in real-time, such that valid strategies can be determined and conducted to avoid traffic congestion [2]. The disaster detecting system (e.g., earthquake, tsunami detection) must deliver the detecting information to the decision making center as soon as possible for disaster prevention [3]. Missing the delay constraints is unacceptable and incurs penalties [4].

Many research efforts have been made to guarantee the stringent delay constraints of bulk data transfers [2]–[5]. They all attempted to carry DS traffic over end-to-end (E2E) connections. A major challenge faced by such E2E transfers is the time- and space-varying nature of background traffic in the inter-datacenter network (inter-DCN) [6]. The bottleneck links could occur on different geographical locations over time. As a result, neither the time window nor the transit bandwidth of the E2E connections can satisfy the requirements of DS traffic. To make matters worse, the demand for DS transfers has been steadily rising, especially at peak hours. To accommodate the peak demand, datacenter (DC) providers must constantly purchase or upgrade expensive link bandwidth even if large amounts of bandwidth remain unused at off-peak hours [7].

Mitigating the peak congestion and improving the link utilization are technically and economically important for DC providers. A promising solution is to introduce DC storage into the data transfer process. Temporarily storing data at intermediate DCs when the next hop is congested and forwarding them at a later time can efficiently reduce the peak-hour demand and improve the bandwidth utilization. This so called Store-and-Forward (SnF) approach has proven to be effective to overcome the E2E challenge [8]. However, the main idea of SnF is to postpone data transmission temporarily, which inevitably incurs the storing delay. Additional delay may be acceptable for delay-tolerant (DT) transfers (e.g., DC backup and data migration), but might be unacceptable for DS transfers. Thus, overcoming the E2E challenge faced by DS transfers remains attractive for DC providers.

In essence, both DT transfer and DS transfer have deadlines for completion time (i.e., the delay constraints). The former has a loose deadline, while the latter has a stringent deadline. As a result, DT transfer can leave a large time scheduling window (given by subtracting the transmission time from the deadline) to perform SnF. In spite of the stringent deadline, SnF has the potential to be used for DS transfer if its time scheduling window is sufficient. Inspired by this, we incorporate the concept of multi-path routing (MP) into SnF, and present an adaptive multi-path SnF scheduling method (AMP-SnF) for DS transfers across the inter-DC optical network.

Our contributions are summarized as follows:

1) AMP-SnF can split DS data into segments and route them through multiple paths, which naturally improves the throughput and accelerates the transfer process. Moreover, the deadline for the entire DS data is consistent with the deadline for each segment. The shorter segments enable wider scheduling windows and hence offer the potential to conduct SnF on each path, which gives extra flexibility in the transfer process.

This work was supported by the National Natural Science Foundation of China (61901118 and 62101372).

978-3-903176-44-7

2) The conventional single-path SnF scheduling problem has proven to be NP-hard [8]. The interplay between data splitting and multi-path selection contributes to a more complex multi-path SnF scheduling (MP-SnF) problem. To simplify the problem, AMP-SnF decouples it into routing problem, SnF-based maximum-flow problem and data splitting problem. AMP-SnF solves them separately.
3) Our study demonstrates that AMP-SnF can not only overcome the E2E challenge and accommodate more DS transfer requests, but also accelerate the transfer process and complete requests earlier. Simulations show that AMP-SnF outperforms the state-of-the-art single-path and multi-path scheduling methods in terms of blocking probability and delivery time. The storing delay incurred by SnF is slight. Thus, AMP-SnF has the potential to improve the user experience of DS services.

The rest of the paper is organized as follows. Sect. II reviews the literature. Sect. III presents AMP-SnF, which is followed by evaluation in Sect. IV. Sect. V concludes this paper.

II. Related Work

A. Existing DS Scheduling Approaches

DS services depend to a great extent on the network's ability to guarantee low E2E delay. Many research efforts have been made to guarantee the stringent delay constraints [2]–[5]. A learning-based proactive resource sharing scheme was proposed to maximize resource utilization efficiency with delay satisfaction for DS services [2]. The work [3] analyzed the low-latency region of a best-effort link and provided strategies to allocate the rate of non-delay-sensitive traffic to the link without negatively affecting the DS traffic. MOTM [4] was presented to jointly determine the computation offloading scheme, the transmission scheduling discipline and the pricing rule for DS services in mobile edge computing. The prior studies [2]–[4] considered single-path routing. On the contrary, a traffic engineering engine, namely DTE-SDN, was proposed to enable large scale scheduling for DS traffic by using MP and dynamic scheduling techniques [5].

All the prior studies considered provisioning DS transfers over the E2E connections. On one hand, they may face the E2E challenge. On the other hand, the DS nature of the transfers prevents the straightforward use of SnF. None of them exploited how to apply SnF to DS transfers.

B. Existing Store-and-Forward Approaches

Bulk data, generated from data migration, disaster recovery and online backups, are often delay-tolerant. The delay tolerance of bulk data hence is exploited. As a forwarding path generally traverses multiple network nodes, DT data can be temporarily stored at intermediate nodes until the next hop is less congested. Taking advantage of the delay tolerance and performing SnF can improve network performance or cost effectiveness [8]. The SnF approaches have been used in inter-DCNs [6], [8], wide-area networks (WANs) [9] and access networks [7]. The prior studies aimed to improve network performance, minimize transfer cost/time and maximize transfer utility by leveraging SnF. During the transfer process, SnF needs to receive the whole data and temporarily stores them until the next hop is available. This incurs significant storing delay and makes it difficult to be used for DS transfers. Therefore, all the prior studies aimed at DT transfers. None of them considered DS transfers.

In [10], a routing framework, namely time-shifted multilayer graph (TS-MLG), was proposed for SnF-enabled optical networks. In this paper, AMP-SnF formulates the MP-SnF problem as multiple maximum-flow (MF) problems by using the TS-MLG, which greatly simplifies the MP-SnF problem.

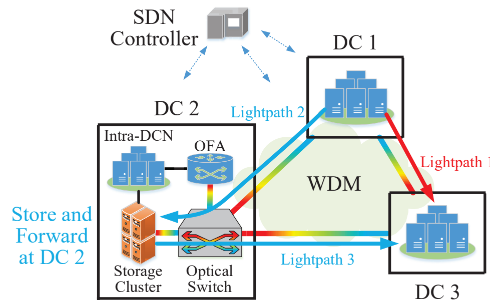


Fig. 1: Illustration of an inter-DC optical network.

III. AMP-SnF: Adaptive Multi-Path 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, as shown in Fig. 1. SDN controller manages optical devices via OpenFlow Agent (OFA). Each DC site can temporarily store bulk data at its storage cluster. In Fig. 1, the storage clusters are used to bypass firewalls and enable high-speed network paths for SnF [7]. Each DC site is capable of wavelength conversion.

Each DS transfer request r is defined by a tuple $r = \{s, d, F, ddl\}$, where s is the source, d is the destination, F is the file size, and ddl is the deadline required to complete r. Assume each request occupies a wavelength for its transmission and each wavelength carries a given data rate. D is defined as the transmission time of r, which is equal to F divided by the data rate of a wavelength. In the context of DS transfers, ddl is assumed to be equal to D. Upon arrival, r is blocked when the transfer completion cannot be guaranteed before the deadline. Let $\boldsymbol{G^L}$ denote a TS-MLG and $\boldsymbol{L}$ denote the layer set in $\boldsymbol{G^L}$.

Compared to the transmission delay, the processing overhead (e.g., scheduling decision and lightpath establishment) is assumed to be negligible. The disk read/write speed of the storage cluster is assumed to be comparable to the transmission capacity of one wavelength. To do so, multiple storage drives need to work in parallel. This is because the read/write rates of each drive are on the order of a few hundred Mbps to a few Gbps depending on the storage techniques, such as hard disk drive (HDD) and solid state drive (SSD). The prior work [7] presented a cost-efficient SSD-based storage architecture for SnF, which is considered in this paper. Assume a portion of the link capacity is dedicated to carrying DS traffic [11].

B. Overview

We incorporate MP into SnF, and present an adaptive multi-path SnF scheduling method for DS transfers, namely AMP-SnF. It orchestrates routing, temporary storage, data splitting and resource allocation for DS requests. The SDN controller maintains a global view of the entire network and all admitted requests. Upon receiving a DS request, AMP-SnF performs admission control in order to determine whether it can be accepted, given the resource usage within its deadline. Once a request is admitted, AMP-SnF guarantees the transfer completion before the deadline.

The key features of AMP-SnF are fourfold:

First, **SnF scheduling with MP capability**. Although SnF can mitigate the E2E issue, it incurs significant storing delay and makes it difficult to be used for DS transfers. On the other hand, MP can improve the throughput and hence be beneficial to DS transfer, but it suffers from the E2E issue. Apparently, MP can be complementary to SnF. AMP-SnF takes advantage of both SnF and MP. It splits DS data into multiple shorter segments and schedules them through multiple paths in a SnF manner. This not only improves the throughput, but also enables more flexibility in the transfer.

Second, **transfer acceleration**. Intuitively, SnF may introduce extra storing delay and hence increase the delivery time. In fact, a shorter segment is more likely to be delivered through either E2E or SnF with less delivery time or storing time. On the other hand, MP improves the throughput. Our studies in Sect. IV show that compared with the conventional single-path scheduling method, AMP-SnF can accelerate DS transfers and shorten the delivery time. Compared with the conventional multi-path scheduling method, AMP-SnF can accommodate more requests at the cost of slightly storing delay.

Third, **adaptive multi-path routing**. In MP, the hop count for each transfer may increase dramatically as the number of paths grows. It requires more bandwidth in total [12]. In some cases, long-hop detours may occur, which, in turn, provokes bandwidth contention and leads to decreased network performance. Besides, the prior work attempted to provision transfers over pre-selected routes [13]. It did not take the dynamics of network status into account. AMP-SnF dynamically calculates a few alternate paths based on the dynamics of network status within the deadline. This not only enables load-balancing routing, but also avoids the detour issue. Moreover, AMP-SnF uses single-path routing as a complement to MP, when MP provisioning fails. Thus, the number of paths used by AMP-SnF can adaptively change with the network status.

Fourth, **flexible data splitting**. In the conventional single-path SnF scheduling problem, both bandwidth and storage resources must be allocated and both spatial routing and temporal scheduling must be performed, which has proven

Algorithm 1 Adaptive Multi-Path SnF Scheduling Method

1: Input: request $r = \{s, d, F, ddl\}$ and the TS-MLG $\boldsymbol{G^L}$
2: Output: $Succ$, $PathSet$, $DataSplit$
3: Initialize: $Succ \leftarrow$False and $PathSet \leftarrow \varnothing$
4: /* Try to provision r using multi-path routing */
5: Create an auxiliary graph G', whose the link weights are updated based on the network status of $\boldsymbol{G^L}$
6: Given ddl, calculate K-link-disjoint-shortest alternate paths from s to d in G' and record the path set as R_{Kdis}
7: Apply Algorithm 2 to find the set of viable paths $PathSet$ from s to d in $\boldsymbol{G^L}$ and the duration assignment $DataSplit$ on each viable path based on R_{Kdis}
8: /* If fail, try to provision r using single-path routing */
9: **if** $Succ$ = False **then**
10:     Calculate K-shortest alternate paths from s to d in the topmost layer of $\boldsymbol{G^L}$ and record the path set as R_K
11:     Apply single-path provisioning algorithm [13] to find a viable path p_i from s to d in $\boldsymbol{G^L}$ based on R_K
12:     $PathSet \leftarrow p_i$ and $DataSplit \leftarrow F$
13: **end if**
14: **if** $Succ$ = True **then**
15:     Accept r, update $\boldsymbol{G^L}$ and OFAs, configure the network based on $PathSet$ and $DataSplit$
16: **else**
17:     Fail to provision and reject r
18: **end if**
19: **return** $Succ$, $PathSet$ and $DataSplit$

to be NP-hard [8]. The interplay between data splitting and multi-path selection results in a more complex MP-SnF problem. To simplify the problem, AMP-SnF decouples it into a routing problem, a SnF-based MF problem and a data splitting problem. (i) Multiple alternate paths are calculated based on the network status. (ii) The MP-SnF problem can be decoupled into multiple single-path SnF scheduling (SP-SnF) problems. The SP-SnF problem is formulated as a SnF-based MF problem using the TS-MLG, which is solved to decide the maximum throughput across this path through SnF. (iii) The data is split into multiple segments, whose sizes are proportional to the maximum throughput of each path. Thus, AMP-SnF can flexibly split data based on the network status.

Algorithm 1 presents the overall procedure of AMP-SnF. AMP-SnF first tries to provision r using multi-path SnF. If AMP-SnF fails, it tries to provision r using single-path routing, where SnF cannot be performed.

Assume that a request r randomly arrives at the network. First, line 5 creates an auxiliary graph G'. Specially, the total leftover wavelengths of each link in the TS-MLG $\boldsymbol{G^L}$ within the time period of ddl add up. The link weight is set to be the reciprocal of the total leftover wavelengths, which would take a key role in load balancing. Second, line 6 applies a K-Disjoint-Paths Algorithm [14] to compute K link-disjoint alternate paths on G' and record the path set as R_{Kdis}. Third, line 7 applies Algorithm 2 to find the set of viable paths

Algorithm 2 Data Splitting and SnF Provisioning Algorithm

1: Input: r, $\boldsymbol{G^L}$, R_{Kdis}
2: Output: $Succ$, $PathSet$, $DataSplit$
3: Initialize: $K' \leftarrow 0$
4: **for** each alternate path $R_i \in R_{Kdis}$ **do**
5:     Create a reduced subgraph G_i'' of $\boldsymbol{G^L}$ based on R_i
6:     Formulate the MP-SnF problem on G_i'' as a MF problem
7:     Calculate $F_{max,i}$ and find p_i for each R_i by solving the MF problem with the algorithms in [10] and [15]
8:     **if** $F_{max,i}$ >0 **then**
9:         Record $F_{max,i}$ and p_i
10:         $K'++$
11:     **end if**
12: **end for**
13: $F_{max,total} \leftarrow \sum_{i \in [1,K']} F_{max,i}$
14: **if** $F_{max,total} \geq F$ **then**
15:     **for** $i = 1; i \leq K'; i++$ **do**
16:         $DataSplit[i] \leftarrow \frac{F_{max,i}}{F_{max,total}} \times F$
17:         $PathSet[i] \leftarrow p_i$
18:     **end for**
19:     **return** $Succ \leftarrow$ True, $PathSet$ and $DataSplit$
20: **else**
21:     **return** $Succ \leftarrow$ False
22: **end if**

$PathSet$ from s to d in $\boldsymbol{G^L}$ through SnF and decide the data splitting $DataSplit$ on each alternate path using R_{Kdis}. Fourth, if Algorithm 2 fails, line 10 calculates K-shortest paths based on the current network status, i.e., the topmost layer of $\boldsymbol{G^L}$ and then, records the path set as R_K. Note that the TS-MLG $\boldsymbol{G^L}$ will be updated dynamically, upon the arrival or departure of requests [10]. Fifth, line 11 applies single-path provisioning algorithm [13] to find p_i using R_K. Sixth, line 12 sets $PathSet$ and $DataSplit$ to be p_i and F respectively, since only a single path would be used. Seventh, line 15 updates $\boldsymbol{G^L}$ and OFAs, configures the network, if r can be accepted. Otherwise, r will be rejected.

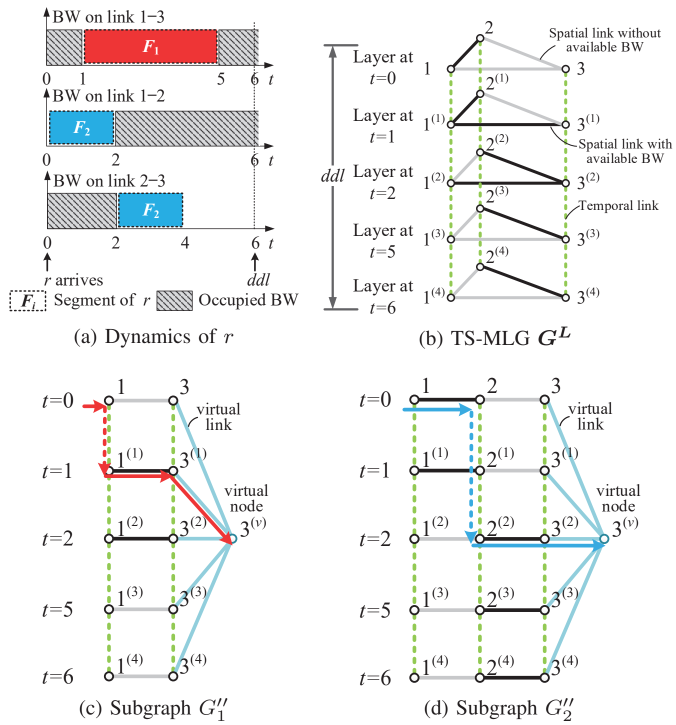
Note that we use the common term "**alternate path**" to refer to a path in a conventional network graph, which simply shows how to reach a destination in the spatial dimension, without considering the temporal dimension. In the TS-MLG, a **viable path** is defined as an alternate path with the required bandwidth/storage resource on each spatial/temporal link.

Algorithm 2 aims to find $PathSet$ and decide $DataSplit$ based on $\boldsymbol{G^L}$. Its main features are threefold:

First, **problem simplification**. The prior SnF scheduling methods involved the status of the entire network in their scheduling problems [8], [9]. Instead, Algorithm 2 only involves the status of an alternate path in scheduling, which simplifies the scheduling problem. Line 5 creates a reduced subgraph G_i'' of $\boldsymbol{G^L}$ based on R_i. Specifically, G_i'' only contains the nodes in R_i and the links that connect these nodes.

Second, **problem formulation**. Line 6 formulates the MP-SnF problem into a SnF-based MF problem on G_i''. In line 7, the algorithm in [10] is used to calculate the maximum throughput of each spatial link in G_i'', which will be used as the weight of the spatial link in G_i''. The weight of temporal link in G_i'' is set to be the residual storage space. The algorithm in [15] is used to solve the MF problem and find a feasible flow through G_i'' that obtains the maximum throughput $F_{max,i}$ and the corresponding viable path p_i for each R_i. Lines 8-11 record $F_{max,i}$ and p_i when $F_{max,i}$ >0.

Third, **data splitting**. Lines 13-22 decide the data splitting $DataSplit[i]$ on each p_i. Line 13 calculates the total throughput of all the paths, i.e., $F_{max,total}$. Lines 15-19 decide $DataSplit[i]$, which is proportional to $F_{max,i}$. Algorithm 2 fails to provision r, if $F_{max,total}$ is smaller than F (line 21).



(a) Dynamics of r (b) TS-MLG $\boldsymbol{G^L}$

(c) Subgraph G_1'' (d) Subgraph G_2''

Fig. 2: Illustration of the provisioning process in AMP-SnF.

C. Example

Here, we use an example to illustrate the basic idea of AMP-SnF in provisioning and accelerating a DS transfer. Consider the inter-DCN shown in Fig. 1. Assume each link has only one wavelength. The occupied wavelength on each link is depicted in Fig. 2(a). For instance, link 1-2 remains free until t=2 and link 2-3 remains busy until t=2. Assume that a request r from DC 1 to DC 3 with F=6 and ddl=6 arrives at t=0. The data rate of a wavelength is assumed to be one unit, the transmission time D hence is equal to 6.

Since none of the links has free wavelengths with the required duration D, r will be rejected when considering a conventional single-path scheduling. Then, considering a single-path scheduling with SnF, the transfer of r has to be postponed until link 1-3 becomes free, i.e., after t=6. Note that the bandwidth gap (from t=1 to t=5) on link 1-3 is insufficient to accommodate r. However, in this case, r fails to meet the deadline.

Fortunately, with AMP-SnF, F can be split into two segments, i.e., F_1=4 and F_2=2. F_1 can be provisioned over link 1-3 (i.e., lightpath 1) at t=1. F_2 can be first provisioned over link 1-2 (i.e., lightpath 2) at t=0 and temporarily stored at DC 2. Once link 2-3 becomes available at t=2, F_2 can be provisioned over lightpath 3. As a result, the transfer of r is completed at t=5, which is earlier than the deadline. Apparently, AMP-SnF not only mitigates the E2E issue, but also accelerates the transfer of r. The dynamics of r is illustrated in Fig. 2(a).

We further show how AMP-SnF is executed with the TS-MLG $\boldsymbol{G^L}$. The corresponding $\boldsymbol{G^L}$ for the wavelength occupation before provisioning r is depicted in Fig. 2(b). $\boldsymbol{G^L}$ consists of 5 layers, with each layer containing 3 nodes. Given an alternate path $R_1 = \{1, 3\}$, $\boldsymbol{G^L}$ is reduced into a subgraph G_1'' depicted in Fig. 2(c). The weight of spatial link in G_1'' is set to be the wavelength idle time using the algorithm in [10]. The weight of temporal link in G_1'' is set to be the residual storage space. Besides, a virtual node of the destination, i.e., $3^{(v)}$, is added to G_1''. Infinite-weight virtual links from 3, $3^{(1)}$,..., $3^{(4)}$ to $3^{(v)}$ are added. Thus, the MP-SnF problem is formulated as a MF problem in G_1''. The algorithm in [15] is executed to solve the MF problem and find a maximum flow from node 1 to node $3^{(v)}$. A viable path depicted in red is found in G_1'', where node 1 is used as temporary storage. Similarly, given an alternate path $R_2 = \{1, 2, 3\}$, a subgraph G_2'' depicted in Fig. 2(d) can be obtained. A viable path depicted in blue is found in G_2'', where node 2 is used as temporary storage.

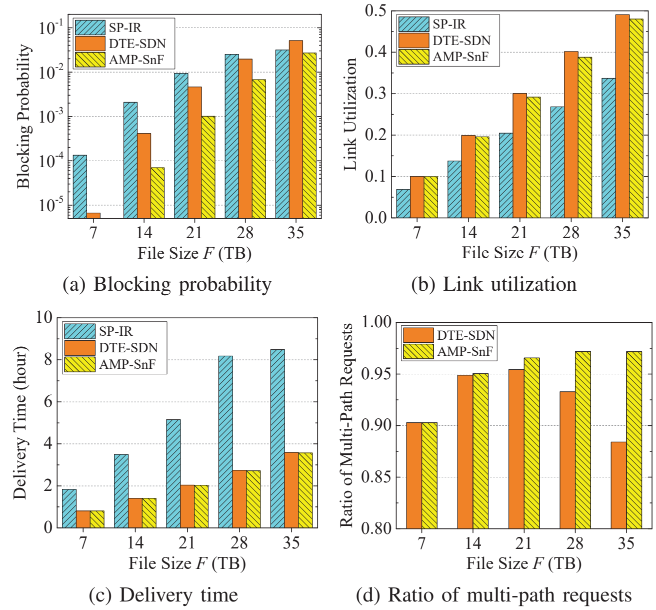
D. Computational Complexity

AMP-SnF decouples the MP-SnF problem into K SP-SnF problems based on each alternate path, which are solved by Algorithm 2 respectively. Thus, the computational complexity of AMP-SnF depends on that of Algorithm 2. Moreover, Algorithm 2 formulates the SP-SnF problem as a MF problem and solves it by using the algorithm [15], whose computational complexity is $O((v \cdot \log e) \cdot (v + e \cdot \log e))$ for the reduced subgraph G_i'' with v nodes and e links. Let N denote the number of nodes along an alternate path R_i and L denote the number of layers in the TS-MLG $\boldsymbol{G^L}$, respectively. In G'', $v = N \cdot L$ and $e = (N-1) \cdot L + (L-1) \cdot N$. The computational complexity of AMP-SnF hence is $O(K \cdot (v \cdot \log e) \cdot (v + e \cdot \log e))$.

IV. Evaluation

We compare AMP-SnF with the state-of-the-art methods in simulations using Matlab. The assumptions in Sect. III-A are adopted. Due to the limited space, a real-world topology (24-node 43-link USNET) is used. Similar results can be obtained in different topologies (e.g., Internet2). Following the assumptions in [1] and [16], five wavelengths on each link are assigned to DS transfers, and each wavelength carries 10 Gbps.

Each DS request randomly selects a source-destination pair from the network. Request arrivals follow a Poisson process with an arrival rate of λ requests per hour. File size for each request follows the negative exponential distribution with an average of F TB. The storage capacity on each DC used for SnF is assumed to be 100 TB. Preliminary results showed that reducing the amount of storage capacity would degrade the performance of SnF. Due to the limited space, the results are omitted. We average the results over 20 simulation runs, each with 500,000 requests.



(a) Blocking probability (b) Link utilization (c) Delivery time (d) Ratio of multi-path requests

Fig. 3: Network performance under various F.

We compare AMP-SnF with the following studies:

1) DTE-SDN [5] is the state-of-the-art multi-path scheduling method. It can adaptively generate multiple E2E paths and distribute the traffic among the paths. Since it is incapable of SnF, it provisions bandwidth immediately, upon receiving a request.
2) SP-IR [13] is a typical single-path scheduling method. It adopts dynamic routing and provisions bandwidth immediately, upon receiving a DS request. SP-IR is also incapable of SnF. A request will be rejected if no E2E lightpaths can be established.

We first investigate how the network changes with F. The maximum number of paths for both AMP-SnF and DTE-SDN is set to be five (i.e., K=5). Following the simulation setting in [5] and [6], λ=6 and $F \in$[7,35] TB. The traffic load can be changed by adjusting either λ or F. Due to the limited space, we increase F in this paper. Similar results can be obtained by increasing λ.

Fig. 3(a) shows that the blocking probability increases with F. When F=7 TB, AMP-SnF obtains a blocking probability of zero. Both AMP-SnF and DTE-SDN outperform SP-IR, since they can distribute the traffic among multiple paths. Benefiting from SnF, AMP-SnF obtains lower blocking probability when compared with DTE-SDN. In Fig. 3(b), both AMP-SnF and DTE-SDN have higher link utilization than SP-IR, since they use more paths for each request. The link utilization in AMP-

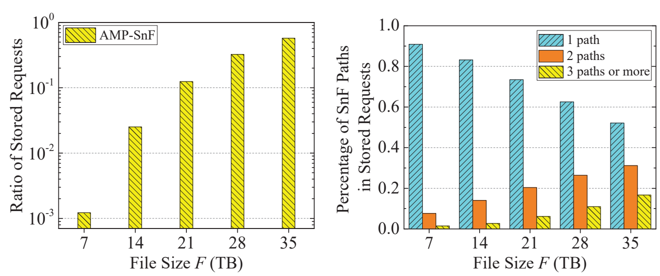
TABLE I: Percentage of the delivery time belonging to the storing delay in the stored requests scheduled by AMP-SnF

File size F (TB)	7	14	21	28	35
Percentage of the delivery time in the stored requests (%)	13.83	14.13	15.65	17.44	19.93

SnF is slightly lower than that in DTE-SDN, because AMP-SnF can use bandwidth more efficiently than DTE-SDN by selecting short-spatial-hop paths via SnF. Delivery time is defined as the time interval between the request arrival instant and the transfer completion instant. In Fig. 3(c), compared with SP-IR, both AMP-SnF and DTE-SDN obtain lower delivery time. This is because MP improves the throughput and hence reduces the delivery time. Furthermore, AMP-SnF obtains the delivery time similar to DTE-SDN. Table I further shows the percentage of the delivery time belonging to the storing delay in the stored requests scheduled by AMP-SnF. This suggests the storing delay introduced by SnF is limited.

Both AMP-SnF and DTE-SDN employ multi-path routing. However, with SnF, AMP-SnF provides extra flexibility in the transfer. To understand that, we further investigate how the ratio of multi-path requests (RMP) changes with F. RMP is defined as the ratio between the number of requests delivered by MP and the number of generated requests. The results are depicted in Fig. 3(d). In DTE-SDN, when F increases beyond 21 TB, the network resources become insufficient to accommodate requests through MP. In this case, requests find it difficult to reserve the required resources on multiple paths simultaneously and hence switch to single-path routing. As a result, the RMP in DTE-SDN greatly decreases when F increases beyond 21 TB. On the contrary, AMP-SnF can leverage bandwidth gaps more efficiently. Requests can still be delivered through multiple paths in AMP-SnF when F increases beyond 21 TB. The RMP in AMP-SnF hence only slightly decreases with F, as shown in Fig. 3(d).

We investigate how AMP-SnF conducts SnF. Ratio of stored requests (RS) is defined as the ratio between the number of the stored requests and the number of generated requests. In Fig. 4(a), the RS in AMP-SnF significantly increases with F. More requests need SnF to reach their destinations when F is medium or higher. Neither SP-IR nor DTE-SDN can perform SnF, their RS results hence are absent. We further



(a) Ratio of stored requests

(b) Percentage of SnF Paths

Fig. 4: Performance of SnF operations in AMP-SnF.

investigate how many paths conduct SnF in the stored requests. A SnF path is defined as a path delivers data through SnF. For example, if a request r distributes data across 3 paths and performs SnF on 2 of 3 paths, the number of SnF paths for r is 2. Fig. 4(b) shows when F=7 TB, less than 1.4% of the stored requests conduct SnF on three paths or more. However, when F increases up to 35 TB, more than 16.7% of the stored requests conduct SnF on three paths or more. The use of SnF becomes more crucial with F growing.

V. Conclusion

In this paper, we present AMP-SnF for DS transfers in the inter-DC optical network. By incorporating MP into SnF, AMP-SnF can provide extra flexibility in provisioning DS transfers. Our studies show that AMP-SnF can ensure the low-latency and high-reliability requirements of DS transfers.

References

[1] L. Chen, S. Liu, and B. Li, "Optimizing network transfers for data analytic jobs across geo-distributed datacenters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 2, pp. 403–414, 2022.

[2] J. Chen, P. Yang, Q. Ye, W. Zhuang, X. Shen, and X. Li, "Learning-based proactive resource allocation for delay-sensitive packet transmission," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 2, pp. 675–688, 2021.

[3] B. Bellalta, "On the low-latency region of best-effort links for delay-sensitive streaming traffic," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 970–974, 2021.

[4] C. Yi, J. Cai, and Z. Su, "A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications," *IEEE Trans. Mob. Comput.*, vol. 19, no. 1, pp. 29–43, 2020.

[5] C. Lin, Y. Bi, H. Zhao, Z. Liu, S. Jia, and J. Zhu, "DTE-SDN: A dynamic traffic engineering engine for delay-sensitive transfer," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5240–5253, 2018.

[6] X. Lin, J. Zou, S. Yue, W. Sun, and W. Hu, "SnF scheduling of multicast transfers across inter-datacenter optical networks," in *IEEE Int. Conf. Commun. (ICC)*, pp. 1–7, 2021.

[7] X. Wang, X. Lin, W. Sun, and M. Veeraraghavan, "Comparison of two sharing modes for a proposed optical enterprise-access SDN architecture," in *Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, pp. 1–8, Nov 2018.

[8] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightwave Technol.*, vol. 35, pp. 5335–5346, Dec. 2017.

[9] C. Lee and J.-K. K. Rhee, "Efficient design and scalable control for store-and-forward capable optical transport networks," *J. Opt. Commun. Netw.*, vol. 9, pp. 699–710, Aug. 2017.

[10] 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," *J. Opt. Commun. Netw.*, vol. 8, pp. 162–174, Mar. 2016.

[11] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations [invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A274–A282, 2013.

[12] Y. Liu, F. Zhou, C. Chen, Z. Zhu, T. Shang, and J.-M. Torres-Moreno, "Disaster protection in inter-datacenter networks leveraging cooperative storage," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 3, pp. 2598–2611, 2021.

[13] W. Lu and Z. Zhu, "IR/AR/MR service provisioning in elastic optical networks," in *Int. Conf. Opt. Commun. Netw. (ICOCN)*, pp. 1–3, 2016.

[14] Q. V. Phung, D. Habibi, H. N. Nguyen, and K. Lo, "K pairs of disjoint paths algorithm for protection in WDM optical networks," in *Asia-Pacific Conf. Commun. (APCC)*, pp. 183–187, 2005.

[15] J. B. Orlin, "A faster strongly polynomial minimum cost flow algorithm," *Oper. Res.*, vol. 41, no. 2, pp. 338–350, 1993.

[16] L. Luo, K. Foerster, S. Schmid, and H. Yu, "Deadline-aware multicast transfers in software-defined optical wide-area networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1584–1599, 2020.