An Edge-Storage-Aided Scheduling Method for Edge-Computing-Enabled Optical Metro Networks

Xiao Lin*, Huihuang Lin*, Shuo Ji*, Hui Qian*, Jun Li†, Shengnan Yue‡
*College of Physics and Information Engineering,
Fuzhou University, Fuzhou, China, {linxiaocer,211127022,201127035,qianhui}@fzu.edu.cn
†School of Electronic and Information Engineering, Soochow University, Suzhou, China, ljun@suda.edu.cn
‡State Key Laboratory of Advanced Optical Communication Systems and Networks,
Shanghai Jiao Tong University, Shanghai, China, shnyue@sjtu.edu.cn

***Abstract*—Along with the rise of edge computing (EC), the cooperation of EC nodes becomes critical, which fuels a growing demand for inter-EC-node (inter-ECN) transfers. Typically, data are transferred in an end-to-end (E2E) manner. However, bulk data, tight transfer deadline and bandwidth fragmentation make such transfers extremely difficult. In this paper, we incorporate multi-path routing and storage on EC nodes into the data transfer, and present an edge-storage-aided multi-path scheduling method (ESMP) for inter-ECN transfers across the EC-enabled optical metro network. ESMP splits the data into chunks and schedules them across link-disjoint routing paths with the help of EC storage. Specifically, within the same deadline for the whole data, a chunk can tolerate longer storing delay and hence can be temporarily stored on an intermediate EC node when its next hop is busy. As a result, a routing path can be split into time-independent sub-paths when the E2E provisioning fails. This improves both the throughput and the flexibility of each transfer. Besides, ESMP can dynamically adapt multi-path routing to balance the completion time and the bandwidth usage based on the current network state. Studies show that compared with the existing scheduling methods, ESMP can accommodate more transfers while reducing the completion time.**

***Index Terms*—Edge computing, multi-path routing, scheduling, storage, optical metro networks.**

I. Introduction

As a promising paradigm, edge computing (EC) brings more computing, storage and networking resources to the edge [1]. Computational tasks from mobile users can be offloaded to nearby EC servers, which guarantees the user's quality of experience (QoE). However, unlike the cloud computing infrastructure, EC servers have limited resources and are located on geo-distributed EC nodes [2]. With the emergence of EC applications, the cooperation of multiple EC nodes is becoming critical [3]. For example, a scheme was developed to tackle the overload problem by offloading the tasks from an overload EC node to nearby light-loaded EC nodes [4]. The work proposed a distributed AI training model over optical metro networks, which distributed training tasks on multiple EC nodes to accelerate the model training [5]. Underlying such edge cooperation, large amounts of data need to be transferred among geo-distributed EC nodes. Such inter-EC-node (inter-ECN) transfers impose a great challenge on the networking infrastructure at the edge.

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

Our study in this paper is mainly motivated by three observations as follows:

First, inter-ECN transfers, generated from the distributed machine learning, Internet of Things (IoT), and the like, often have tight deadlines [3]. In other words, although the inter-ECN transfers do not need to start immediately like typical delay-sensitive transfers, they have hard and tight deadlines. Missing the deadline would be unacceptable for some EC applications [6]. Moreover, it is very beneficial to complete the transfers as early as possible. Many efforts hence have been made to accelerate the transfers by using multi-path routing (MP) [7]–[9]. For each transfer, the more the routing paths, the higher the throughput and the shorter the completion time. The bandwidth usage for each transfer significantly grows with the number of routing paths increasing [7]. However, the networking bandwidth at the edge is limited and expensive [10]. Bandwidth contention and network congestion may escalate when MP is used inappropriately. This, in turn, degrades the network performance (e.g., blocking probability) [7]. Thus, how to balance the trade-off between the completion time and the network performance should be carefully considered in MP.

Second, the background traffic varies widely in both space and time [10]. For example, a campus/enterprise network often peaks during the work hours, whereas an ADSL/FTTH access network often peaks in the late evening. Such spatiotemporal variations in the background traffic result in an undesirable phenomenon known as bandwidth fragmentation. As a result, the existing efforts find it difficult to fully utilize the bandwidth fragments across times and locations, since they use end-to-end (E2E) connections to deliver data. To accommodate the growing peak demand, EC operators have to upgrade or purchase expensive link bandwidth constantly even if the average utilization is quite low [11]. Thus, how to efficiently use the bandwidth fragments to accommodate inter-ECN transfers is technically and economically important for EC operators.

Third, EC servers typically have a certain amount of storage resources. EC storage was often used for the task computation like CPU and memory [1]–[5]. This naturally inspires us to explore the potential of introducing EC storage into the data transfer process and using it like networking resources.

Our prior work [12] and [13] aimed to accelerate data transfers or reduce the transfer completion time by evenly splitting data through K link-disjoint paths. However, our studies showed that using MP may degrade the network performance when the traffic load was high and the network resources are insufficient [12]. Therefore, there is a trade-off between the completion time and the bandwidth usage, which needs flexible adaption based on the current network state [13]. In this paper, we aim to tackle this issue and present an edge-storage-aided multi-path scheduling method (ESMP) for inter-ECN transfers. Our contributions are summarized as follows:

1) The combination of EC storage and MP has mutual benefits. On one hand, ESMP splits data into small chunks before sending them. This equivalently enables larger time windows to schedule chunks (e.g., temporary storage), since the deadline for each chunk remains constant, regardless of how the data are split. On the other hand, by temporarily storing chunks at intermediate nodes, ESMP splits each E2E path into time-independent sub-paths when it fails to provision an entire path. This not only overcomes the E2E challenge, but also efficiently utilizes the residual bandwidth.
2) Inspired by the trade-off, the routing and data splitting problem is formulated as an optimization model, which aims to dynamically balance the completion time and the total bandwidth usage for each transfer based on the network state. ESMP hence can dynamically decide the number of paths and the chunk size for each path.
3) Studies show that ESMP can efficiently utilize the bandwidth to accommodate more transfers and accelerate the transfers. Compared with the existing scheduling methods, ESMP has lower blocking probability than the existing methods. This comes at the cost of slightly higher completion time when the traffic load is medium or higher. ESMP has the potential to meet the requirement of inter-ECN transfers.

The rest of the paper is organized as follows. Sect. II presents ESMP, which is followed by evaluation in Sect. III. Sect. IV concludes this paper.

II. ESMP: Edge-Storage-Aided Multi-Path Scheduling Method

A. Network Model and Assumptions

The infrastructure of the optical metro network is a wavelength-division multiplexing (WDM) network, as depicted in Fig. 1. A typical EC node can be deployed in the central office or the aggregation point of the optical metro network and is equipped with an EC server, an OpenFlow switch and an optical switch. The EC node can temporarily store data at its EC server. The optical switch has wavelength converters. A SDN controller maintains a global view of the EC servers and optical devices in a centralized manner.

A 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. Requests randomly arrive the network with a Poisson process and λ requests per second. File size is exponentially distributed with an average of F GB. Each request occupies a wavelength for its transmission. The transmission time D is equal to F divided by the data rate of a wavelength. Let K denote the maximum number of paths available for ESMP. Intermediate EC nodes will not perform data splitting/reassembly. Only the source/destination performs data splitting/reassembly. The overhead incurred by data splitting/reassembly is assumed to be negligible [14].

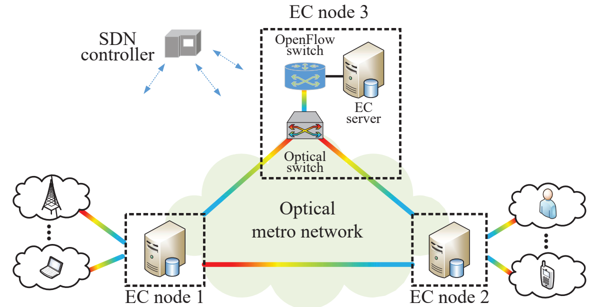

Fig. 1: The illustration of the network scenario.

B. Overview

We present ESMP to schedule the inter-ECN transfers across the EC-enabled optical metro network. The main idea of ESMP is to split data into small chunks and route them across multiple paths with the help of EC storage on intermediate nodes. Its key features are fourfold as follows:

First, **storage-aided multi-path routing**. EC storage and MP are complementary to each other in ESMP. On one hand, for each path, a chunk can be temporarily stored on an intermediate EC node when its next hop is busy and forwarded at a later time. With intermediate storage nodes, a conventional E2E path can be split into time-independent sub-paths. In other words, ESMP can leverage EC storage to bring together the bandwidth fragments across different times and locations. This not only improves the flexibility of transfers, but also efficiently utilizes the bandwidth fragments. On the other hand, the small chunks need less transmission time and hence tolerate longer storing delay within the same deadline. This suggests a large time window to temporarily schedule the transfer of each chunk. Compared with delivering the whole data using single-path routing (SP), the multiple chunks are more likely to be delivered by leveraging intermediate storage.

Second, **deadline-aware scheduling**. Intuitively, the use of EC storage introduces extra storing delay into the transfer. ESMP will consider the deadline constraint as well as the storing delay introduced by each chunk in the scheduling process. A request will be admitted only when ESMP can ensure all the chunks reach the destination before the deadline.

Third, **dynamic balance between the completion time and the total bandwidth usage**. As previously discussed, there exists a trade-off between the completion time and the total bandwidth usage in MP. The objective of ESMP is to dynamically balance this trade-off. Specifically, ESMP jointly minimizes the weighted sum of the completion time and the total bandwidth usage. The weights are dynamically adjusted. When the network resources are sufficient, ESMP intends to use more paths for each transfer in order to minimize the completion time at the cost of using more bandwidth. Otherwise, ESMP intends to use fewer paths for each transfer in order to save the bandwidth resources to accommodate more transfers at the cost of longer completion time.

Fourth, **efficient problem decoupling**. In the conventional MP, the interplay between routing and data splitting often requires a complex optimization model for decision making [7]. The use of EC storage results in a more complex storage-aided multi-path scheduling (SA-MP) problem, which involves data splitting, multi-path routing, storage node selection, temporal scheduling and bandwidth/storage resource allocation. Since the bandwidth fragments can be distributed across different times and locations, the size of the SA-MP problem dramatically increases with the time window of temporal scheduling as well as the network topology expanding. This contributes to a substantial computational cost. To simplify the problem efficiently, ESMP decouples it into a routing and data splitting sub-problem and multiple storage-aided single-path scheduling (SA-SP) sub-problems. i) The first sub-problem is solved to jointly decide the alternate routes and the chunk assigned to each alternate route. ii) Based on the optimal results, the SA-MP problem can be decoupled into multiple SA-SP problems. Each SA-SP problem is solved to schedule transmission/storage and allocate resources on a given route.

C. Problem Formulation

Network operators use the link utilization as a measure of the entire network resource usage. In practice, data are often delivered over only a small number of paths [7]. Thus, we introduce the related link utilization ratio, i.e., α, as a measure of the wavelength usage along the alternate routes from s to d. Let $R_{(s,d)}$ denote the set of the pre-computed K-link-disjoint routes from s to d. Here, α is defined as the ratio of the number of busy wavelengths and the total number of wavelengths on all links of $R_{(s,d)}$. We have

$$\alpha = \frac{\sum\limits_{i,j\in R_{(s,d)}} w_{(i,j)}^{busy}}{\sum\limits_{i,j\in R_{(s,d)}} w_{(i,j)}} \tag{1}$$

where $w_{(i,j)}^{busy}$ denotes the number of busy wavelengths on link (i,j) and $w_{(i,j)}$ denotes the total number of wavelengths on link (i,j). The routing and data splitting problem in ESMP is formulated as an optimization model.

Given:

- $G(V,E)$: the network topology, where V is the set of optical nodes, E is the set of fiber links connecting nodes in V, $\{i|i\in V\}$, $\{j|j\in V\}$ and $\{(i,j)|(i,j)\in E\}$;
- $F_{(i,j)}^{max}$: the maximum throughput of link (i,j);
- $H_{(s,d)}$: the hop count of the shortest route in $R_{(s,d)}$.

Variables:

- $x_{(i,j)}$: integer variable, denotes a chunk using link (i,j);
- x^{max}: integer variable, denotes the largest chunk;
- $f_{(i,j)}$: binary variable, equals one if link (i,j) is used, and zero otherwise.

Objective:

$$\begin{aligned} &\min\ \alpha \times \frac{\sum\limits_{i,j\in V} f_{(i,j)}}{H_{(s,d)}} + (1-\alpha)\times\frac{x^{max}}{F} \\ &s.t. \quad \text{Constraints}\,(3)-(9). \end{aligned} \tag{2}$$

In the objective, i.e., Eq. (2), the first term calculates the total hop count of all the routing paths, and the second term denotes the largest chunk. To compare the two terms properly, the first term is normalized by $H_{(s,d)}$ and the second term is normalized by F. The value of α is dynamically calculated based on the current wavelength usage along $R_{(s,d)}$. The wavelength resources along $R_{(s,d)}$ are sufficient, when α is small. In this case, ESMP intends to minimize the size of the largest chunk by using more paths. As a result, the transfer is more likely to have an early completion at the cost of using more paths and hence using more wavelength resources. On the contrary, the wavelength resources are insufficient, when α is large. In this case, ESMP intends to minimize the hop count of all the routing paths and hence save more wavelength resources for ongoing transfers. As a result, more transfers would be accommodated at the cost of longer completion time.

Constraints:

1) *Flow conservation constraints:* Eq. (3)-Eq. (5) ensure the flow conservation of the source, the destination and the intermediate nodes, respectively.

$$\sum_{i\in V} x_{(s,i)} - \sum_{i\in V} x_{(i,s)} = F \tag{3}$$

$$\sum_{i\in V} x_{(d,i)} - \sum_{i\in V} x_{(i,d)} = -F \tag{4}$$

$$\sum_{i\in V} x_{(j,i)} - \sum_{i\in V} x_{(i,j)} = 0,\ \forall j\in V, j\neq s, j\neq d \tag{5}$$

2) *Data splitting constraints:* Eq. (6) ensures that each link is assigned to only a chunk and its size is within $F_{(i,j)}^{max}$. Eq. (7) finds the largest chunk.

$$f_{(i,j)} \le x_{(i,j)} \le F_{(i,j)}^{max}\times f_{(i,j)},\ \forall i,j\in V \tag{6}$$

$$x^{max} = \max(x_{(i,j)}),\ \forall i,j\in V \tag{7}$$

3) *Multi-path routing constraints:* Eq. (8) ensures the number of paths is within K. Eq. (9) ensures data splitting and reassembly only occur on s and d, respectively.

$$\sum_{i\in V} f_{(s,i)} \le K,\ \sum_{i\in V} f_{(i,d)} \le K \tag{8}$$

$$\sum_{j\in V} f_{(i,j)} \le 1,\ \forall i\in V, i\neq s, i\neq d \tag{9}$$

Algorithm 1 Edge-Storage-Aided Multi-Path Scheduling Method (ESMP)

1: Input: request $r = \{s, d, F, ddl\}$, the TS-MLG $\boldsymbol{G^L}$
2: Output: $PathSet$ and $ChunkSet$
3: Initialize: $PathSet \leftarrow \varnothing$ and $Chunks \leftarrow \varnothing$
4: Apply the algorithm in [15] to find the maximum throughput $F^{max}_{(i,j)}$ of link (i,j) within ddl in $\boldsymbol{G^L}$
5: Create an auxiliary graph G using $F^{max}_{(i,j)}$ as link cost
6: Calculate α using Eq. (1)
7: Formulate the routing and data splitting problem into the optimization model given in Sect. II-C
8: Solve the optimization model to obtain the alternate route set $RouteSet$ and the chunk set $ChunkSet$, where $R_i \in RouteSet$ and $C_i \in ChunkSet$
9: **for all** $R_i \in RouteSet$ **do**
10:     Create a reduced graph G_i of $\boldsymbol{G^L}$ based on R_i
11:     Formulate the SA-SP problem into a routing problem on G_i
12:     Apply Algorithm 2 to solve the problem, which returns a viable path p_i
13:     **if** C_i can be accommodated on R_i **then**
14:         $PathSet \leftarrow PathSet \cup p_i$
15:         $Succ \leftarrow$ True
16:     **else**
17:         $Succ \leftarrow$ False and Break
18:     **end if**
19: **end for**
20: **if** $Succ$ = True **then**
21:     Update $\boldsymbol{G^L}$ and reconfigure the network based on $PathSet$ and $ChunkSet$
22: **else**
23:     Reject r, $PathSet \leftarrow \varnothing$ and $ChunkSet \leftarrow \varnothing$
24: **end if**
25: **return** $PathSet$ and $ChunkSet$

Algorithm 2 Storage-Aided Single-Path Scheduling (SA-SP) Algorithm

1: Input: r, G_i and C_i
2: Output: p_i
3: Initialize: $p_i \leftarrow \varnothing$
4: Apply the algorithm in [15] to find the available throughput of each link
5: Create a multilayer graph G'_i whose link cost is one if the available throughput is sufficient for C_i, and zero otherwise
6: **for all** layer $l_j \in G'_i$ **do**
7:     Apply the BFS algorithm on G'_i to find a viable path p_i from s at layer l_1 to d at layer l_j
8:     **if** a viable path p_i is found **then**
9:         $Succ \leftarrow$ True and Break
10:     **else**
11:         $Succ \leftarrow$ False and Continue
12:     **end if**
13: **end for**
14: **return** p_i

D. Algorithm

The procedure of ESMP is illustrated in Algorithm 1. Since ESMP takes the spatiotemporal-varying bandwidth fragments into account, it employs a time-shifted multilayer graph (TS-MLG) [15] to capture the dynamics of the network state. In the TS-MLG, each layer is a snapshot of the network at a certain time. ESMP uses the TS-MLG $\boldsymbol{G^L}$ to manage the state information of the entire network, where L denotes the number of layers in the TS-MLG.

Assume that a request r arrives at time t. First, line 4 applies the algorithm in [15] to $\boldsymbol{G^L}$ in order to decide the maximum throughput $F^{max}_{(i,j)}$ of link (i,j) within the time period of ddl. Second, line 5 creates an auxiliary graph G and $F^{max}_{(i,j)}$ is used as the link cost for link (i,j) in G. Third, line 6 calculates α using Eq. (1) and the required state information can be obtained from the topmost layer of $\boldsymbol{G^L}$. Fourth, lines 7-8 solve the optimization model given in Sect. II-C using a commercial solver Gurobi [16] to decide the alternate routes and the chunk assigned to each route. Let $RouteSet$ denote the set of alternate routes and $ChunkSet$ denote the set of chunks, where a route $R_i \in RouteSet$ and a chunk $C_i \in ChunkSet$. Fifth, lines 9-19 decouple the SA-MP problem into multiple SA-SP problems and solve them separately. Lines 10-11 create a reduced subgraph G_i of $\boldsymbol{G^L}$ based on R_i and formulate the SA-SP problem as a routing problem on G_i, which will be elaborated upon in the following subsection. Line 12 solves each SA-SP problem using Algorithm 2 to decide storage nodes, temporal scheduling and resource allocation, which returns a viable path p_i. If all the chunks can be accommodated, r will be accepted. The network will be reconfigured and $\boldsymbol{G^L}$ will be updated in line 21. Otherwise, line 23 rejects r.

The concepts of spatial link and temporal link are introduced in the TS-MLG [15]. When data traverse a spatial link, it suggests transiting data from one node to another. When data traverse a temporal link, it suggests storing data on a node for a certain period. When applying a routing algorithm on the TS-MLG, both transmission and storage can be jointly scheduled. Thus, by using the TS-MLG, the SA-SP problem can be formulated as a conventional routing problem. Algorithm 2 aims to solve the SA-SP problem by using the Breadth First Search (BFS) algorithm to find a viable path p_i on G_i. Line 4-5 create a multilayer graph G'_i based on G_i, whose link cost is one if the throughput of a link is sufficient for C_i, and zero otherwise. Line 6-13 attempt to find a viable path from the source d at layer l_1 to the destination d at layer l_j repeatedly using the BFS algorithm. If a viable path p_i is found, line 14 returns p_i. Otherwise, line 14 returns a null set for p_i.

E. Example

We show how ESMP schedules a request r across a four-node network in Fig. 2. The link capacity is assumed to be one wavelength. The TS-MLG $\boldsymbol{G^L}$ captures the dynamics of wavelength usage, as depicted in Fig. 2(a). For example, link

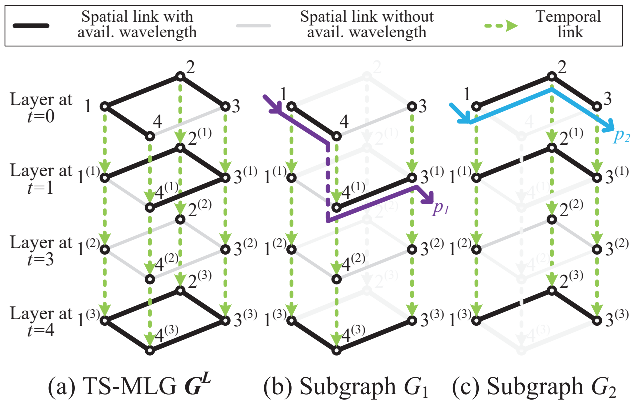

Fig. 2: Illustration of the provisioning process in ESMP.

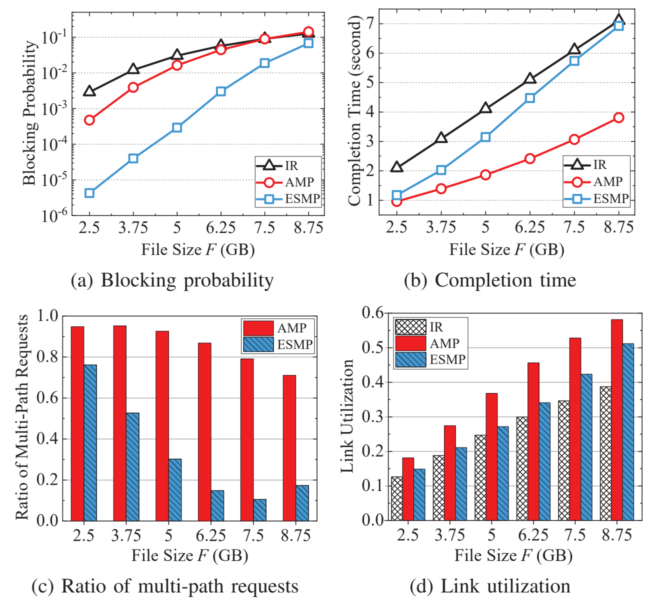

Fig. 3: Network performance under various F.

(4,3) is busy at t=0 and becomes available at t=1. Consider a request r, where s=1, d=3, F=4 and ddl=4. It arrives at t=0. For simplicity, the data rate of a wavelength is one unit. Thus, D=4.

There are two paths from s to d. The E2E route R_1=$\{1, 4, 3\}$ is unavailable at t=0. Although the E2E route R_2=$\{1, 2, 3\}$ is available at t=0, its holding time is 3 and insufficient for r. As a result, neither SP nor MP is able to provision r.

Fortunately, r can be provisioned by ESMP. By solving the optimization model, the data F are split into two chunks, i.e., C_1=1 and C_2=3, assigned to R_1 and R_2, respectively. ESMP decouples the SA-MP problem into two SA-SP problems based on R_1 and R_2, and solves them separately using Algorithm 2. For the SA-SP problem on R_1, ESMP first reduces $\boldsymbol{G^L}$ into a subgraph G_1 using R_1, as shown in Fig. 2(b). In G_1, the available throughput of each link is calculated by using the algorithm in [15], which attempts to find the throughput of a link within ddl. The weight of this link is set to be one if the available throughput is sufficient for the chunk size and zero otherwise. As such, the SA-SP problem is formulated as routing problems. By executing the BFS algorithm on G_1 and G_2, viable paths p_1 and p_2 are found, as depicted in Figs. 2(b) and (c). In G_1, C_1 is transmitted and stored on node 2, and then re-transmitted to node 3. C_1 completes at t=2. C_2 starts to transmit immediately and completes at t=3. In summary, the transfers of all the chunks are completed before ddl.

Compared with SP and MP, ESMP not only accommodates r, but also accelerates r. From the network's perspective, it is desirable to admit as many transfers as possible and complete them as early as possible. Thus, ESMP has the potential to meet the transfer requirements of EC applications.

F. Computational Complexity

The complexity of ESMP depends on the optimization model and Algorithm 2. As the model has $O(|E|)$ variables and $O(|E| + |V|)$ constraints, its complexity is $O((|E| + |V|)^2 \cdot |E|)$. Algorithm 2 uses the BFS algorithm to search the subgraph G_i, whose complexity is $O(|R_i| \cdot |L|)$. Since the maximum of paths is K, the complexity of ESMP hence is $O((|E| + |V|)^2 \cdot |E| + K \cdot |R_i| \cdot |L|)$. In some cases, the optimization model may be too large to handle. Future direction of our study is to simplify the optimization model.

III. Evaluation

In this section, ESMP is compared with the existing methods in Matlab simulations. We use a 38-node and 59-link metro network [4]. The assumptions in Sect. II-A are adopted. A portion of the link bandwidth (50 Gbps) are dedicated for inter-ECN traffic. The data rate of a wavelength is 10 Gbps [17]. The source and destination of a request spread evenly across all nodes in the network. The storage capacity of each EC server is set to be 100 GB. Intuitively, the higher the storage capacity, the better performance ESMP can achieve. How to dimension the storage capacity is worthy of further study. The simulation results are averaged over 15 runs. In each run, 1,000,000 requests are generated.

We compare ESMP with an adaptive MP scheduling method (AMP) [18] and a dynamic SP scheduling method (IR) [19]. Both AMP and IR cannot store data on intermediate EC nodes. In this case, a request will be admitted only when E2E connections are able to be established immediately, upon its arrival. With EC storage, ESMP is more flexible than AMP and IR. We consider a scenario, where ddl=D and the deadlines are too tight to tolerate any storing delay if the whole data are delivered by SP. In this case, ESMP cannot perform temporary storage when the data are delivered over a single forwarding path. We investigate how the performance of the three methods perform under different values of F. Herein, K=5, λ=10 and F∈[2,9] GB.

Fig. 3(a) shows that ESMP obtains the lowest blocking probability among the three methods. Benefiting from EC storage and MP, ESMP is more flexible and hence accommodates more requests than AMP and IR. Compared with IR,

TABLE I: The stored requests in ESMP

File size F (GB)	2.5	5	7.5
Ratio of stored requests (%)	0.36	1.83	14.11
Storing delay (second)	0.21	0.58	1.57
Completion time (second)	1.41	3.09	5.64

AMP obtains lower blocking probability when F is low and medium. When F increases beyond 7 GB, AMP suffers from higher blocking probability than IR, because AMP intends to use more paths for each request. The network is more likely to become insufficient to accommodate ongoing requests, especially when F is medium or higher. On the contrary, ESMP can dynamically balance the completion time and the bandwidth usage, it intends to use fewer paths and save more resources for ongoing requests when F is medium or higher. In Fig. 3(b), AMP has shorter completion time than ESMP and IR, but it suffers from higher blocking probability when F is high. Like AMP, ESMP has shorter completion time when F is low. However, its completion time gradually increases with F in order to obtain lower blocking probability.

We investigate how many requests would use MP, i.e., multi-path requests. Fig. 3(c) shows the ratio of multi-path requests in AMP and ESMP. Although both ESMP and AMP leverage MP, ESMP can adapt the number of routing paths more efficiently. The ratio of multi-path requests in ESMP dramatically decreases when F grows, as shown in Fig. 3(c). Fig. 3(d) shows that the link utilization in ESMP is lower than that in AMP, but higher than that in IR. This suggests that compared with AMP and IR, ESMP can use wavelengths more efficiently in order to balance the trade-off.

Then, we investigate how ESMP leverages EC storage. As seen in Table I, the larger the value of F, the higher the ratio of stored requests, the more the requests need temporary storage. This implies that EC storage is becoming critical in inter-ECN transfers. However, the use of storage introduces extra storing delay, which grows with F. For example, the storing delay contributes to 15% of the completion time when F=2.5 GB. This number increases up to 28% when F=7.5 GB. When F is medium or higher, the inter-ECN transfers have to experience longer storing delay.

IV. Conclusion

In this paper, ESMP is presented to schedule data transfers through the optical metro network. By combining EC storage with adaptive MP, ESMP can not only improve the throughput and accelerate the transfer, but also split the E2E path into time-independent sub-paths and utilize the spatiotemporal bandwidth fragments. In addition, ESMP can dynamically adapt the routing paths and the chunk sizes to balance the trade-off between the completion time and the bandwidth usage based on the current network state. Our studies show that ESMP outperforms IR in terms of blocking probability and completion time. Compared with AMP, ESMP obtains lower blocking probability and similar completion time when the traffic load is light. When the traffic load increases, ESMP can accommodate more requests than AMP by proactively using fewer routing paths.

References

[1] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan, and E. Hossain, "Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, pp. 1–26, 2021.

[2] I. AlQerm and J. Pan, "Enhanced online Q-learning scheme for resource allocation with maximum utility and fairness in edge-IoT networks," *IEEE Trans. Netw. Sci. and Eng.*, vol. 7, no. 4, pp. 3074–3086, 2020.

[3] H. Wang, Z. Qu, Q. Zhou, H. Zhang, B. Luo, W. Xu, S. Guo, and R. Li, "A comprehensive survey on training acceleration for large machine learning models in iot," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 939–963, 2022.

[4] S. Huang, C. Yang, S. Yin, Z. Zhang, and Y. Chu, "Latency-aware task peer offloading on overloaded server in multi-access edge computing system interconnected by metro optical networks," *J. Lightwave Technol.*, vol. 38, no. 21, pp. 5949–5961, 2020.

[5] Y. Li, Z. Zeng, J. Li, B. Yan, Y. Zhao, and J. Zhang, "Distributed model training based on data parallelism in edge computing-enabled elastic optical networks," *IEEE Commun. Lett.*, pp. 1–1, 2020.

[6] X. Lin, Y. Li, J. Shao, and Y. Li, "Storage-assisted optical upstream transport scheme for task offloading in multi-access edge computing," *J. Opt. Commun. Netw.*, vol. 14, no. 3, pp. 140–152, 2022.

[7] 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.

[8] S. Cai, F. Zhou, Z. Zhang, and A. Meddahi, "Disaster-resilient service function chain embedding based on multi-path routing," in *IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, pp. 1–7, 2021.

[9] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, "Software defined optical networks (SDONs): A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 4, pp. 2738–2786, 2016.

[10] J. Ma, H. Yang, B. Bao, Q. Yao, T. Peng, H. Guo, and J. zhang, "Tidal traffic prediction model of metropolitan optical network based on long short-term memory," in *Asia Commun. Photonics Conf. (ACP)*, pp. 1–3, 2020.

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

[12] X. Lin, S. Ji, S. Yue, J. Li, W. Sun, and W. Hu, "Adaptive multi-path snf scheduling method for delay-sensitive transfers across inter-datacenter optical networks," in *Int. Conf. Opt. Netw. Des. Model. (ONDM)*, pp. 1–6, 2022.

[13] X. Lin, S. Ji, S. Yue, J. Li, and F. Chen, "Storage-enabled adaptive multi-path scheduling method for data transfers in task offloading," in *IEEE Conf. High Perform. Switch. Rout. (HPSR)*, pp. 238–244, 2022.

[14] A. A. Atallah, G. B. Hamad, and O. A. Mohamed, "Routing and scheduling of time-triggered traffic in time-sensitive networks," *IEEE Trans. Industr. Inform.*, vol. 16, no. 7, pp. 4525–4534, 2020.

[15] 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.

[16] "Gurobi Optimization." http://www.gurobi.com, 2022.

[17] 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.

[18] 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.

[19] 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.