Journal of Network and Systems Management (2019) 27:166–187
https://doi.org/10.1007/s10922-018-9463-2

Optimizing Trade-Off Between Cost and Performance of Data Transfers Using Bandwidth Reservation in Dedicated Networks

Liudong Zuo[1] · **Michelle M. Zhu**[2] · **Chia-Han Chang**[1]

Received: 17 April 2017 / Revised: 20 June 2018 / Accepted: 24 June 2018 / Published online: 5 July 2018

Abstract

Because of the solid performance of providing quality of service for various applications for decades, bandwidth reservation has been increasingly used in recent years for large amounts of data transfer to achieve guaranteed performance. However, effective scheduling strategy to achieve the trade-off between data transfer cost and data transfer performance still remains to be investigated. In this paper, we focus on the trade-off between cost and the most common performance parameter, i.e., completion time, of data transfers using bandwidth reservation in dedicated networks. We consider the scheduling of two types of bandwidth reservation requests regarding such trade-off: (1) to achieve the minimum data transfer cost given the data transfer deadline, and (2) to achieve the earliest data transfer completion time given the maximum data transfer cost. We propose two bandwidth reservation algorithms with rigorous optimality proofs to optimize the scheduling of these two types of bandwidth reservation requests. We then compare the proposed algorithms with two scheduling algorithms originating from one widely used scheduling algorithm in production networks, and the efficacy of the proposed optimal algorithms is verified through extensive simulations.

Keywords Bandwidth reservation/scheduling · Dynamic provisioning · High-performance networks · QoS

✉ Liudong Zuo
lzuo@csudh.edu

Michelle M. Zhu
zhumi@montclair.edu

Chia-Han Chang
cchang71@toromail.csudh.edu

[1] Computer Science Department, California State University Dominguez Hills, Carson, CA 90747, USA

[2] Department of Computer Science, Montclair State University, Montclair, NJ 07043, USA

1 Introduction

In the late twentieth century, bandwidth reservation strategy was designed to provide quality of service (QoS) for real-time multimedia applications, such as satellite communication and video-conferencing. A number of bandwidth reservation protocols and models were then proposed, such as resource reservation protocol (RSVP) [1], asynchronous transfer mode (ATM) [2] and internet integrated service model [3, 4]. Because of its solid performance, bandwidth reservation has been increasingly used in recent years for the transfer of extremely large amounts of data [5–9]. For example, the Large Hadron Collider (LHC), the most well-known high-energy particle accelerator, can generate up to 30 petabytes of data per year [10], and the climate data in climate science is expected to exceed 100 exabytes by 2020 [11]. Such sheer volume of data is normally generated at one data center and then needs to be transferred to other geographically distributed data centers for collaboration [8, 12]. Bandwidth reservation on dedicated channels of high-performance networks (HPNs) has proven very effective for such extremely large amounts of data transfer. For example, the data generated by LHC has been transferred globally using the on-demand secure circuits and advance reservation system (OSCARS) deployed in the energy sciences network (ESnet) [13], one of the most widely used bandwidth reservation services in scientific area.

Besides ESnet, there are many other similar HPNs providing bandwidth reservation services, such as Internet2 ION [14], circuit-switched high-speed end-to-end transport architecture [15], user controlled light paths [16], Japanese Gigabit Network II [17], UltraScience Net [18] and Bandwidth on Demand in Geant2 network [19]. Considering the exponential growth of the data generated from the next generation scientific research applications, the bandwidth reservation service provided by dedicated HPNs is expected to be deployed and used by more and more applications across the globe.

To make the data transfers using bandwidth reservation, bandwidth reservation requests (BRRs) are firstly created, specifying properties of the data transfers as well as the scheduling constraints and performance requirements. One typical BRR specifies the following data property parameters: the source end-site, the destination end-site, size of the data to be transferred, the maximum local area network (LAN) bandwidth, the data available time, and the data transfer deadline. After the receipt of one BRR, the underlying scheduling network then employs scheduling algorithms to identify the data transfer path and allocate bandwidth resources on that path within a certain time interval. Only when all the constraints and requirements of the received BRR have been successfully satisfied can the corresponding data transfer start.

Although many different problems regarding bandwidth reservation have been studied in the past decades, there are still some critical problems to be investigated. One of them is the design of effective scheduling strategy to achieve the trade-off between data transfer cost and other data transfer performance parameters. Challenges of this problem arise from the requirements desired by both

the users and the bandwidth reservation service providers. As for the users, the most common data transfer performance parameter is the data transfer completion time. However, many times the users require their data transfers to be finished not only before the specific deadlines, but also with the minimal costs charged by the bandwidth reservation service providers under certain data transfer cost model. While for the bandwidth reservation service providers, successfully transferring the data using the minimum bandwidth resources is highly desired to achieve high system throughput for the maximum profit and resource utilization.

In this paper, we focus on the trade-off between data transfer cost and the most common data transfer performance, i.e., data transfer completion time. We consider the scheduling of two types of BRRs regarding such trade-off: (1) to achieve the minimum data transfer cost given the data transfer deadline, referred to as MinC-TC (minimize data transfer cost with time constraint), and (2) to achieve the earliest data transfer completion time given the maximum data transfer cost, referred to MinT-CC (minimize data transfer completion time with cost constraint). We assume that the data transfer cost is mainly determined by the amount of the reserved bandwidth, time duration of the bandwidth reservation and length of the bandwidth reservation path. We propose two bandwidth reservation algorithms with rigorous optimality proofs, called Opt-MinC-TC and Opt-MinT-CC ("Opt" denotes "Optimal"), to optimize the scheduling of these two types of BRRs. We then compare the proposed algorithms with two scheduling algorithms originating from one widely used scheduling algorithm in production networks, called FBR-MinC-TC and FBR-MinT-CC ("FBR" denotes "Flexible Bandwidth Reservation"), from the perspective of various performance metrics. The efficacy of Opt-MinC-TC and Opt-MinT-CC is verified through extensive simulations on simulated ESnet.

The rest of this paper is organized as follows. We show the work related to bandwidth reservation in dedicated HPNs in Sect. 2. The mathematical models, concepts of bandwidth reservation and problem formulation are presented in Sect. 3. Detailed algorithm designs and illustrations of Opt-MinC-TC and FBR-MinC-TC, and Opt-MinT-CC and FBR-MinT-CC are given in Sects. 4 and 5, respectively. We conduct extensive simulations and results analysis in Sect. 6, and conclude our work in Sect. 7.

2 Related Work

Because of the wide use of bandwidth reservation service for the immense data transfer to achieve the guaranteed performance, various problems have been investigated in the past few years. To show these related work in a clearer way, we tabulate them in Table 1. For each existing research shown in Table 1, if complexity of the studied problem is NP-complete, the corresponding NP-complete proof is given and heuristic algorithm is then proposed; otherwise, the optimal algorithm is designed and the corresponding optimality proof is given except [9]. Compared with existing work, our problems under investigation are very different and unique from both the perspectives of objectives and algorithm design.

Table 1 Bandwidth reservation related work in recent years

Author(s) and publication year(s)	Problem(s) studied	Problem(s) complexity
Zuo et al. [5]	Given multiple BRRs in a batch awaiting to be scheduled in a network, following two scheduling maximization problems were studied: (1) maximize the amount of data to be transferred, and (2) maximize the number of requests to be scheduled	NP-complete
Zuo et al. [20]	If the given BRR cannot be optimally scheduled in a scheduling network, two alternative reservation options are computed and returned to the user to choose: schedule the BRR within the closest time intervals before and after the user-specified time interval. Two different types of BRRs were considered: one focuses on bandwidth and the other focuses on data transfer	P
Wu [6]	Two advance scheduling problems in overlay networks with linear capacity constraints were studied: Fixed-Bandwidth Path and Varying-Bandwidth Path, with the objective to achieve the earliest data transfer completion time for a given data size	NP-complete
Zuo [20, 21]	Given a BRR with higher priority, the bandwidth preemption on one link of the scheduling network was studied following two different constraints: (1) minimize the number and then the total bandwidth of existing bandwidth reservations to be preempted, and (2) minimize the total bandwidth and then the number of existing bandwidth reservations to be preempted	NP-complete
Zuo et al. [7]	Given a BRR to be scheduled in a scheduling network, following two problems regarding data transfer reliability were studied: (1) achieve the highest data transfer reliability under a given data transfer deadline, and (2) achieve the earliest data transfer completion time under a given minimum data transfer reliability	P
Wang et al. [22, 23]	Maximize the number of satisfied requests for fixed bandwidth reservation [22] and variable slot-bandwidth reservation [23] with deadline constraint on a fixed network path	NP-complete
Hou et al. [24]	A generic problem of bandwidth scheduling with two variable node-disjoint paths was studied. Two variable paths of fixed or variable bandwidth with negligible or non-negligible switching delay were further considered	NP-complete
Zuo et al. [25, 26]	Schedule all BRRs within a batch while achieving their best average earliest completion time and shortest duration in a network [12] and on one fixed network path [25, 26]	NP-complete
Zuo et al. [10, 27]	Given a batch of BRRs with different priorities, optimally schedule each BRR with the earliest data transfer completion time and that with the shortest data transfer duration	P

Table 1 (continued)

Author(s) and publication year(s)	Problem(s) studied	Problem(s) complexity
Lin and Wu [8]	An exhaustive combination of the path and bandwidth constraints resulted in four different types of advance bandwidth scheduling problems with the objective to achieve the earliest data transfer completion time given a BRR: (1) fixed path with fixed bandwidth, (2) fixed path with variable bandwidth, (3) variable path with fixed bandwidth, and (4) variable path with variable bandwidth	NP-complete or P
Balman [9]	Given a BRR, identify the bandwidth reservation option with the earliest completion time and that with the shortest duration in a network	P

3 Mathematical Models, Concepts of Bandwidth Reservation and Problem Formulation

In this section, we first show mathematical models and concepts of bandwidth reservation, followed by data transfer cost model and problem formulation.

3.1 Mathematical Models and Concepts of Bandwidth Reservation

A number of definitions and parameters will be introduced in this section to show the mathematical models and concepts of bandwidth reservation in a concise and clearer way. For convenience, several parameters are tabulated in Table 2.

The two types of BRRs considered in the paper, namely MinC-TC and MinT-CC, are described as follows:

- $(v_s, v_d, B^{max}, D, [t^S, t^E])$: Identify the bandwidth reservation option with the minimum data transfer cost under the constraint that the data transfer must be completed no later than the preset deadline t^E;
- $(v_s, v_d, B^{max}, D, t^S, C^{max})$: Identify the bandwidth reservation option with the earliest data transfer completion time under the constraint that the data transfer cost must not exceed the preset maximum cost C^{max}.

In the above notations, D denotes size of the data to be transferred from the source node v_s to the destination node v_d, while t^S and B^{max} denote the earliest

Table 2 Definitions of some parameters introduced in Sect. 3

Parameters	Definitions
v_s	Source node
v_d	Destination node
B^{max}	Maximum LAN bandwidth constraint
D	Size of data to be transferred
t^S	Earliest data transfer start time
t^E	Latest data transfer end time (deadline)
C^{max}	Maximum data transfer cost
p	Reservation path/data transfer path
$L(p)$	Length of path p
b	Reserved bandwidth
t^s	Data transfer start time
t^e	Data transfer end time
ts	Time step
tw	Time window
$B(e, ts)$	Available bandwidth of edge e within ts
$B(p, tw)$	Available bandwidth of path p within tw

possible data transfer start time and the maximum LAN bandwidth constraint, respectively [10]. The reserved bandwidth for one BRR is upper limited by B^{max}.

We model a scheduling network as a graph $G(V, E)$, where V and E represent the set of nodes and edges, respectively. Suppose we have an example scheduling network G shown on the left side of Fig. 1, we have $V = \{v_s, v_1, v_2, v_d\}$ and $E = \{v_s - v_1, v_s - v_2, v_1 - v_2, v_2 - v_d\}$. The available bandwidth table showing the available bandwidth of each edge within time interval $[0, \infty)$ is presented on the right side of Fig. 1. Bandwidth capacity of an edge is defined as the amount of available bandwidth of that edge without any bandwidth reservation on it. We suppose the bandwidth capacity of each edge of G is its maximum available bandwidth within time interval [0, 11 s] and there is no bandwidth reservation on any edge of G after $11s$. We suppose G has one MinC-TC BRR to schedule, and the BRR requires to transfer 36 Gb data from v_s to v_d within time interval [0, 11 s] with the minimum data transfer cost. Suppose the specified maximum LAN bandwidth of the received BRR is 12 Gb/s, we can represent the above MinC-TC BRR as $(v_s, v_d, 12\,\text{Gb/s}, 36\,\text{Gb}, [0, 11\,\text{s}])$.

As we can see from Fig. 1, available bandwidth of an edge might change from time to time. Such bandwidth dynamism is caused by the dynamic bandwidth reservation and release on the edge [12]. Given time interval $[t^S, t^E]$ and scheduling network G, for any time point $t \in [t^S, t^E]$, if any edge of G has different available bandwidths at time point $t - \delta$ and $t + \delta$, $\delta \to 0$, we call time point t a time dot [5]. These two end points of the given interval $[t^S, t^E]$, t^S and t^E, are also regarded as time dots. For example, G shown in Fig. 1 has three time dots within [0, 11 s], i.e., {0, 6 s, 11 s}, and four time dots within $[0, \infty)$, i.e., $\{0, 6\,\text{s}, 11\,\text{s}, \infty\}$.

Given time dots list $\{td_0, td_1, \ldots, td_{n-1}\}$, the time interval between any two adjacent time dots is called a time step while the time interval between any two different time dots is called a time window. Hence, a time step has the form $[td_i, td_{i+1}]$, $0 \le i \le n-2$, while a time window has the form $[td_i, td_j]$, $0 \le i < j \le n-1$. In the

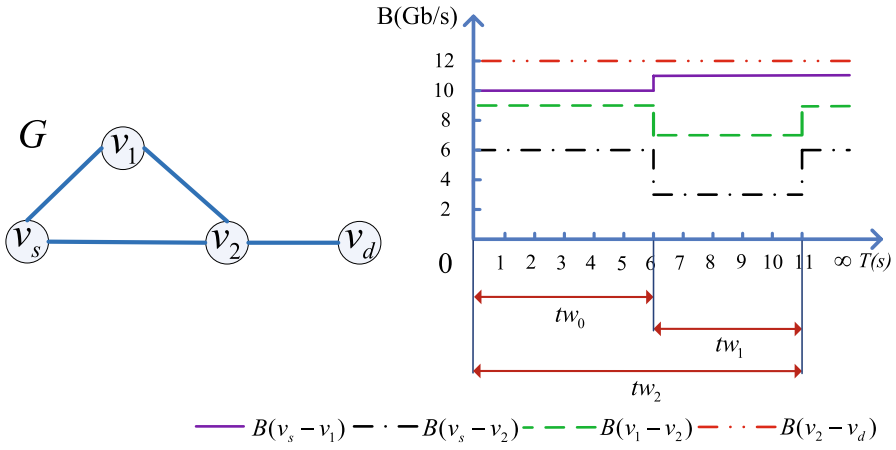

Fig. 1 Topology of an example scheduling network (left) and the available bandwidth table of each edge within time interval $[0, \infty)$ (right)

rest of the paper, we also denote time step i as $ts_i = [ts_i^s, ts_i^e]$ and time window i as $tw_i = [tw_i^s, tw_i^e]$. For example, within [0, 11 s], G shown in Fig. 1 has two time steps, i.e., $ts_0 = [0, 6\,\text{s}]$ and $ts_1 = [6\,\text{s}, 11\,\text{s}]$, and three time windows, i.e., $tw_0 = [0, 6\,\text{s}]$, $tw_1 = [6\,\text{s}, 11\,\text{s}]$ and $tw_2 = [0, 11\,\text{s}]$ (a time step is also a time window). These three time windows are shown in Fig. 1. It is not difficult to see that the available bandwidths of all edges of G within a time step do not change, and a time window consists of one time step or multiple consecutive time steps.

We represent the available bandwidth of edge e within time step ts_i as $B(e, ts_i)$, which can be directly read from the available bandwidth table of G. For example, within time step $ts_0 = [0, 6\,\text{s}]$, we have $B(v_s - v_1, ts_0) = 10\,\text{Gb/s}$. The bandwidth resource of an edge within a time step is defined as the maximum amount of data that edge can transfer within that time step. So for edge e within time step $ts_i = [ts_i^s, ts_i^e]$, its bandwidth resource equals $B(e, ts_i) \cdot (ts_i^e - ts_i^s)$. For example, for edge $v_s - v_1$ within time step $ts_0 = [0, 6\,\text{s}]$, its bandwidth resource equals $10\,\text{Gb/s} \cdot (6\,\text{s} - 0) = 60\,\text{Gb}$. The bandwidth resource of G within time interval $[t^S, t^E]$ equals sum of the bandwidth resources of all edges within all time steps contained in $[t^S, t^E]$.

Available bandwidth of edge e within time window tw_i equals the smallest available bandwidth of e among all time steps contained in tw_i. For example, time window tw_2 consists of two time steps $ts_0 = [0, 6\,\text{s}]$ and $ts_1 = [6\,\text{s}, 11\,\text{s}]$, so available bandwidth of edge $v_s - v_1$ within time window tw_2 equals $min(10\,\text{Gb/s}, 11\,\text{Gb/s}) = 10\,\text{Gb/s}$. Similarly, available bandwidth of path p within time window tw_i, denoted by $B(p, tw_i)$, is limited by the bottleneck edge of p, namely, the edge with the minimum available bandwidth within tw_i. For example, path $v_s - v_2 - v_d$ consists of two edges $v_s - v_2$ and $v_2 - v_d$, and available bandwidths of these two edges within time window tw_2 are 3 Gb/s and 12 Gb/s, respectively, so $B(v_s - v_2 - v_d, tw_2) = min(3\,\text{Gb/s}, 12\,\text{Gb/s}) = 3\,\text{Gb/s}$. We use $L(p)$ to denote length of path p, namely the number of edges p contains.

For a given BRR, we say it can be successfully scheduled if we can make a bandwidth reservation option on one path of G and this option can satisfy all the requirements and constraints of that BRR. For convenience, we call the above path a qualified path and the reservation option a qualified reservation (QR), denoted by $(p, b, [t^s, t^e], c)$, where p, b, t^s, t^e and c denote the qualified path (also the data transfer path), the reserved bandwidth on the qualified path, data transfer start time, data transfer end time, and data transfer cost, respectively [10]. A path within a time window is called a qualified path only when we can make at least one QR on that path within that time window. Given a BRR, we normally can identify multiple qualified paths and make multiple QRs on these paths within different time windows. For example, for the given MinC-TC BRR $(v_s, v_d, 12\,\text{Gb/s}, 36\,\text{Gb}, [0, 11\,\text{s}])$, we can make infinite QRs for it on paths $v_s - v_2 - v_d$ and $v_s - v_1 - v_2 - v_d$ of G within time interval [0, 11 s]. Here we are interested in two special QRs: the one with the minimum data transfer cost, denoted as QRMC, and the one with the earliest completion time, denoted as QRECT. Focus of this paper is design of the scheduling algorithms to identify QRMC for the given MinC-TC BRR and QRECT for the given MinT-CC BRR.

Given a BRR, if we want to successfully schedule it within tw_i, length of tw_i should be at least $\frac{D}{B^{max}}$, i.e., $tw_i^e - tw_i^s \geq \frac{D}{B^{max}}$, and the reserved bandwidth within tw_i should be at least $\frac{D}{tw_i^e - tw_i^s}$. We further derive that we can remove those edges with available bandwidths less than $\frac{D}{tw_i^e - tw_i^s}$ within tw_i, and this pruning does not affect the scheduling feasibility of the given BRR within tw_i. For example, for the given MinC-TC BRR ($v_s, v_d, 12\,\text{Gb/s}, 36\,\text{Gb}, [0, 11\,\text{s}]$) within time window [6 s, 11 s], the minimum bandwidth is $\frac{36\,\text{Gb}}{(11\,\text{S}-6\,\text{S})} = 7.2\,\text{Gb/s}$. As we can see from Fig. 1, neither edge $v_1 - v_2$ nor $v_s - v_2$ has adequate available bandwidth, so both edges can be removed when we try to schedule the given BRR within [6 s, 11 s]. Such redundant edges pruning strategy gives us two benefits: (1) the efficiency of the data transfer path identification process can be greatly improved because of the reduced searching space using Dijkstra's algorithm, and (2) the non-*NULL* path returned by Dijkstra's algorithm can finish the data transfer of the corresponding BRR for sure, which improves the BRR scheduling ratio. Such benefits will be further explained in Sects. 4–6.

3.2 Data Transfer Cost Model and Problem Formulation

The total bandwidth resource consumed by a QR is defined as the total bandwidth resource consumed to transfer data of the corresponding BRR, which equals

$$L(p) \cdot b \cdot (t^e - t^s) = L(p) \cdot D, \tag{1}$$

where D is size of the data to be transferred in the corresponding BRR.

For the bandwidth reservation service providers, the profit comes from transferring data of the BRRs from users. In this paper, we assume the profit of providing bandwidth reservation service for a BRR within a time interval, namely the data transfer cost of that BRR, equals the overall amount of bandwidth resource consumed by the BRR to transfer its data within that time interval. With this data transfer cost model, it is easy to see from Eq. 1 that the best case, namely the minimum data transfer cost, for one BRR is that length of the data transfer path is 1, namely the source node and the destination node are directly connected by one edge.

Based on our analysis, we further describe these two types of BRRs, MinC-TC and MinT-CC, as follows:

- $(v_s, v_d, B^{max}, D, [t^S, t^E])$: Identify the qualified path with the least length among all possible qualified paths within all possible time windows contained in $[t^S, t^E]$, and return the corresponding QR on it, which is the QRMC of the BRR;
- $(v_s, v_d, B^{max}, D, t^S, C^{max})$: Identify the qualified path satisfying the following two constraints, and return the corresponding QR on it, which is the QRECT of the BRR: (1) its length is no larger than $\lfloor \frac{C^{max}}{D} \rfloor$, and (2) data transfer on it has the earliest completion time among all possible qualified paths within all possible time windows contained in $[t^S, \infty)$.

Now scheduling these two types of BRRs comes down to how to identify the optimal paths.

4 Algorithm Design and Analysis for MinC-TC

In this section, we focus on the algorithm design and analysis for MinC-TC. Its optimal algorithm, Opt-MinC-TC, is proposed followed by the comparison algorithm, FBR-MinC-TC. For each algorithm, we present the detailed algorithm design and brief explanation, and then illustrate it using an example.

4.1 Algorithm Design and Analysis of Opt-MinC-TC

4.1.1 Algorithm Design and Explanation

As mentioned in the data transfer cost model in Sect. 3.2, data transfer cost of one BRR is actually proportional to the length of the data transfer path, so to achieve the minimum data transfer cost, we should make the bandwidth reservation on the path with the least length from the source node to the destination node within a certain time interval. Please refer to Algorithm 1 for the detailed algorithm design and pseudocode of Opt-MinC-TC. Its complexity is $O\big(|td|^2 \cdot (|E| + |V| \cdot \log |V|)\big)$ in the worst case. In the algorithm, we use a list named *ltw* to store the time windows within time interval $[t^S, t^E]$. Opt-MinC-TC is briefly explained as follows:

Line 5–6: If length of time window $[td_i, td_j]$ is less than $\frac{D}{B^{max}}$, we know that the given MinC-TC BRR cannot be successfully scheduled within it. Hence, we only need to consider those time windows with the lengths at least $\frac{D}{B^{max}}$.

Line 7: We consider the scheduling of the given MinC-TC BRR within each time window in *ltw*.

Lines 8–14: Within time window $tw_i \in ltw$, we remove those edges with available bandwidth less than $\frac{D}{tw_i^e - tw_i^s}$. Such pruning technique is explained in Sect. 3. After the pruning, if remaining of G becomes disconnected, and v_s and v_d are in two different components, we continue to the next time window in *ltw* because no path in current time window has enough available bandwidth to finish the data transfer of the given BRR; otherwise, we employ Dijkstra's algorithm to identify the path with the least length from v_s to v_d. We use variable p to record the qualified path with the least length among all time windows in *ltw*, and *tw* to record the corresponding time window (Line 14).

Line 15–18: After the iteration of all time windows in *ltw*, $p \neq NULL$ denotes that we could successfully identify the qualified path with the least length. We then make and return the QR on p within time window *tw* as shown in Line 16, which is the QRMC of the given BRR. While after the iteration of *ltw*, $p == NULL$ denotes we could not identify any qualified path within any time

window in *ltw*, the given BRR could not be successfully scheduled, we then return *NULL*.

Algorithm 1 Opt-MinC-TC

GIVEN: $G(V, E)$
INPUT: MinC-TC BRR $(v_s, v_d, B^{max}, D, [t^S, t^E])$
OUTPUT: QRMC or $NULL$ if no QR can be identified

1: Initialize path $p \leftarrow NULL$, $L(p) \leftarrow \infty$, time window $tw \leftarrow NULL$, and time window list $ltw \leftarrow NULL$;
2: Identify all time dots of G within time interval $[t^S, t^E]$ (including t^S and t^E), and then put them into TreeSet td in ascending order;
3: **for** $i \leftarrow 0$ to $|td| - 2$ **do**
4: **for** $j \leftarrow i + 1$ to $|td| - 1$ **do**
5: **if** $td_j - td_i \geq \frac{D}{B^{max}}$ **then**
6: Add time window $[td_i, td_j]$ into time window list ltw;
7: **for** $i \leftarrow 0$ to $|ltw| - 1$ **do**
8: Remove those edges with available bandwidths less than $\frac{D}{tw_i^e - tw_i^s}$. Denote the current network topology as G';
9: **if** G' becomes disconnected, and v_s and v_d are in two different components **then**
10: Continue;
11: **else**
12: Run Dijkstra's algorithm to identify the path with the least length from v_s to v_d. Denote the returned path as p';
13: **if** $p' \neq NULL$ && $L(p') < L(p)$ **then**
14: $p = p'$ and $tw = tw_i$;
15: **if** $p \neq NULL$ **then**
16: Return $\left(p, min\left(B^{max}, B(p, tw)\right), tw^s, tw^s + \frac{D}{min(B^{max}, B(p,tw))}, D \cdot L(p)\right)$.
17: **else**
18: Return $NULL$.

4.1.2 Algorithm Illustration

We illustrate Opt-MinC-TC using the example scheduling network G shown in Fig. 1 and the example MinC-TC BRR (v_s, v_d, 12 Gb/s, 36 Gb, [0, 11 s]).

Initialize parameters as shown in Line 1 of Algorithm 1. For the example network G, the identified time window list is $ltw = \{[0, 6\,\text{s}], [0, 11\,\text{s}], [6\,\text{s}, 11\,\text{s}]\}$. Iterate through *ltw*. Within time window [0, 6 s], we try to remove those edges with available bandwidths less than $\frac{36\,\text{Gb}}{6\,\text{s}-0} = 6\,\text{Gb/s}$, and no edge will be removed. The path with the least length from v_s to v_d is $v_s - v_2 - v_d$ with available bandwidth of 6 Gb/s and length of 2. Within time window [0, 11 s], we use similar strategy and edge $v_s - v_2$ will be removed, the path with the least length from v_s to v_d is $v_s - v_1 - v_2 - v_d$ with available bandwidth of 7 Gb/s and length of 3. Within time window [6 s, 11 s], we use similar strategy, and edges $v_s - v_2$ and $v_1 - v_2$ will be removed. After the removal, G' becomes disconnected, and v_s and v_d are in two different components, time window iteration stops here. We have $p = v_s - v_2 - v_d$ and $tw = [0, 6\,\text{s}]$. Following Line 16 of Algorithm 1, the corresponding QRMC is $(v_s - v_2 - v_d, 6\,\text{Gb/s}, [0, 6\,\text{s}], 72)$ and will be returned.

4.1.3 Optimality Proof

Theorem 1 *Opt-MinC-TC returns the QRMC, if it exists, for the input MinC-TC BRR.*

Proof We use proof-by-contradiction to prove the theorem. Suppose for the input BRR, its QRMC exists and is made on path p'' within time window tw''. As we will see, the QR identified using Opt-MinC-TC is not *NULL*, and we suppose this QR is made on path p and we have $L(p'') < L(p)$, namely length of the data transfer path of the QR identified by Opt-MinC-TC is larger than that of the optimal data transfer path.

Let us consider the scheduling within tw'' using Opt-MinC-TC. From our supposition, we know that p'' can finish the data transfer of the input BRR. So after the edge pruning procedure stated in Line 8 of Algorithm 1, v_s and v_d are in the same component of G'. Line 12 of Algorithm 1 denotes the path with the least length from v_s to v_d is p', hence we have $L(p'') \geq L(p')$. During the time window iteration, we use variable p to record the qualified path with the least length among all qualified paths within all time windows in ltw, so after the time window iteration, we have $L(p) \leq L(p')$. Along with $L(p'') \geq L(p')$, we have $L(p'') \geq L(p)$, which contradicts our initial supposition that $L(p'') < L(p)$. Proof ends. □

4.2 Algorithm Design and Analysis of FBR-MinC-TC

4.2.1 Algorithm Design and Explanation

Design of FBR-MinC-TC originates from the scheduling algorithm proposed by the scientists at Lawrence Berkeley National Laboratory, who is currently managing ESnet [9]. Please refer to Algorithm 2 for the detailed algorithm design and pseudocode of FBR-MinC-TC. In the worst case, its complexity is also $O\big(|td|^2 \cdot (|E| + |V| \cdot \log |V|)\big)$. Line 3 of Algorithm 2 denotes that within each time window in ltw, we directly employ Dijkstra's algorithm to identify the path with the least length from v_s to v_d.

Algorithm 2 FBR-MinC-TC

GIVEN: $G(V, E)$
INPUT: MinC-TC BRR $(v_s, v_d, B^{max}, D, [t^S, t^E])$
OUTPUT: Estimated QRMC or $NULL$ if no QR can be identified

1: The same as Lines 1 – 6 of Algorithm 1;
2: **for** $i \leftarrow 0$ to $|ltw| - 1$ **do**
3: Run Dijkstra's algorithm to identify the path with the least length p' from v_s to v_d;
4: **if** $p' \neq NULL$ && $\left(tw_i^e - tw_i^s\right)) \cdot min\left(B^{max}, B(p', tw_i)\right) \geq D$ && $L(p') < L(p)$ **then**
5: $p = p'$ and $tw = tw_i$;
6: The same as Lines 15 – 18 of Algorithm 1.

4.2.2 Algorithm Illustration

We illustrate FBR-MinC-TC using G shown in Fig. 1, and the example MinC-TC BRR (v_s, v_d, 12 Gb/s, 36 Gb, [0, 11 s]).

Among all three time windows in $ltw = \{[0, 6\,\text{s}], [0, 11\,\text{s}], [6\,\text{s}, 11\,\text{s}]\}$, the path with the least length from v_s and v_d is the same as $v_s - v_2 - v_d$. After time window iteration, the estimated QRMC will be made on the recorded path $v_s - v_2 - v_d$ within recorded time window [0, 6 s]: ($v_s - v_2 - v_d$, 6 Gb/s, [0, 6 s], 72).

5 Algorithm Design and Analysis for MinT-CC

In this section, we focus on the algorithm design and analysis for MinT-CC. Its optimal algorithm, Opt-MinT-CC, is proposed followed by the comparison algorithm, FBR-MinT-CC. For each algorithm, we present the detailed algorithm design and brief explanation, and then illustrate it using an example.

5.1 Algorithm Design and Analysis of Opt-MinT-CC

5.1.1 Algorithm Design and Explanation

Please refer to Algorithm 3 for the detailed algorithm design and pseudocode of Opt-MinT-CC. Its complexity is $O\big(|td|^2 \cdot |E| \cdot (|E| + |V| \cdot \log |V|)\big)$ in the worst case. Opt-MinT-CC is briefly explained as follows:

Lines 9–14: After the edge pruning process (Line 5), if v_s and v_d are in the same component of G', we do the operations stated in Line 9. Every time we add one edge to G', we run Dijkstra's algorithm and try to identify the path with the least length from v_s to v_d through the newly added edge. If the returned path $p' \neq NULL$ and its length is no larger than $\lfloor \frac{C^{max}}{D} \rfloor$, then the required data transfer could be successfully finished on p' and the cost is no larger than the preset maximum cost C^{max}. If the data transfer on path p' has earlier completion time than t, Line 14 is executed and the edge iteration (Line 10) stops here. Because the edges in E' are sorted by their available bandwidth in descending order, the other qualified paths, if there are, within current time window will not have more available bandwidth than p', namely the data transfers on these qualified paths will not have an earlier completion time. So it is not necessary to consider these qualified paths.

Lines 15–18: During the time window iteration, we use variable t to record the earliest data transfer completion time. After the time window iteration, $p \neq NULL$ denotes that we can successfully identify at least one qualified path, we then make and return the corresponding QR on p following Line 16.

Algorithm 3 Opt-MinT-CC

GIVEN: $G(V, E)$
INPUT: MinT-CC BRR $(v_s, v_d, B^{max}, D, t^S, C^{max})$
OUTPUT: QRECT or $NULL$ if no QR can be identified

1: Initialize path $p \leftarrow NULL$, $t \leftarrow \infty$, time window $tw \leftarrow NULL$, and time window list $ltw \leftarrow NULL$;
2: Identify all time dots of G within time interval $[t^S, \infty)$ (including t^S and ∞), and then put them into TreeSet td in ascending order;
3: The same as Lines 3 – 6 of Algorithm 1;
4: **for** $i \leftarrow 0$ to $|ltw| - 1$ **do**
5: Remove those edges with available bandwidths less than $\frac{D}{tw_i^e - tw_i^s}$. Denote the current network topology as G';
6: **if** G' becomes disconnected, and v_s and v_d are in two different components **then**
7: Continue;
8: **else**
9: Denote the edge set of G' as E'. Sort edges in E' by their available bandwidths in descending order. Remove all edges from G';
10: **for** $e \in E'$ **do**
11: Add e to G', and then run Dijkstra's algorithm to identify the path with the least length p' from v_s to v_d through e;
12: **if** $p' \neq NULL$ && $L(p') \leq \lfloor \frac{C^{max}}{D} \rfloor$ && $tw_i^s + \frac{D}{min(B^{max}, B(p', tw_i))} < t$ **then**
13: $t = tw_i^s + \frac{D}{min(B^{max}, B(p', tw_i))}$, $p = p'$ and $tw = tw_i$;
14: Break;
15: **if** $p \neq NULL$ **then**
16: Return $\left(p, min\left(B^{max}, B(p, tw)\right), tw^s, t, D \cdot L(p)\right)$.
17: **else**
18: Return $NULL$.

5.1.2 Algorithm Illustration

We illustrate Opt-MinT-CC using the example scheduling network G shown in Fig. 1 and the example MinT-CC BRR $(v_s, v_d, 12\,\text{Gb/s}, 54\,\text{Gb}, 0, 200)$.

Initialize parameters as shown in Line 1 of Algorithm 3, and the identified time window list is $ltw = \{[0, 6\,\text{s}], [0, 11\,\text{s}], [0, \infty), [6\,\text{s}, 11\,\text{s}], [6\,\text{s}, \infty), [11\,\text{s}, \infty)\}$. Iterate through ltw. Within time window $[0, 6\,\text{s}]$, edge $v_s - v_2$ will be removed. The sorted remaining edge set becomes $E' = \{v_2 - v_d, v_s - v_1, v_1 - v_2\}$. Remove all edges from G', and then add edges in E' to G' one at a time. The first path returned by Dijkstra's algorithm is $v_s - v_1 - v_2 - v_d$ with available bandwidth of 9 Gb/s and length of 3. We have $3 \leq \lfloor \frac{200}{54} \rfloor = 3$, and $0 + \frac{54\,\text{Gb}}{9\,\text{Gb/s}} = 6\,\text{s}$. We then have $t = 6\,\text{s}$, $p = v_s - v_1 - v_2 - v_d$ and $tw = [0, 6\,\text{s}]$, and the edge adding loop stops here. Within time window $[0, 11\,\text{s}]$, using similar strategy, we know that path $v_s - v_1 - v_2 - v_d$ is also the first path returned by Dijkstra's algorithm. However, completion time of the data transfer on that path is $\frac{54\,\text{Gb}}{7\,\text{Gb/s}} \approx 7.71\,\text{s} > t = 6\,\text{s}$, so the time window iteration continues. Similarly, we could not identify any QR with an earlier data transfer completion time within rest of the time windows. After the time window iteration, we have $p = v_s - v_1 - v_2 - v_d$ and $tw = [0, 6\,\text{s}]$, so the corresponding QRECT is $(v_s - v_1 - v_2 - v_d, 9\,\text{Gb/s}, [0, 6\,\text{s}], 162)$ and will be returned.

5.1.3 Optimality Proof

Theorem 2 *Opt-MinT-CC returns the QRECT, if it exists, for the input MinT-CC BRR.*

Proof We use proof-by-contradiction to prove the theorem. Suppose for the input BRR, its QRECT exists and is made on path p'' within time window tw'' with the data transfer completion time of t''. As we will see, the QR identified using Opt-MinT-CC is not *NULL*, and we suppose this QR is made on path p with the data transfer completion time of t. We assume $t'' < t$.

Let us consider the scheduling process within time window tw''. The edge pruning procedure will not remove any edge on path p'' since it could finish the data transfer of the given BRR. We suppose the bottleneck edge of path p'' is edge e'', then we have $e'' \in E'$. We have two cases for the edge adding loop (Line 10 of Algorithm 3): The loop stops before or after edge e'' is added. We consider each case as follows:

Case (1): The edge adding loop stops before edge e'' is added. Since $e'' \in E'$, this case happens when the Break statement on Line 14 of Algorithm 3 is executed and we have found a qualified path p' before adding e'' to G'. Since all edges in E' is sorted by their available bandwidths in descending order and p' is found before adding e'' to G', we have $B(p', tw'') \geq B(e'', tw'') = B(p'', tw'')$. Then we have $tw''^s + \frac{D}{min(B^{max}, B(p', tw''))} \leq tw''^s + \frac{D}{min(B^{max}, B(p'', tw''))} = t''$. As we can see from Lines 12 – 13 of Algorithm 3, we use path p to denote the qualified path with the earliest data transfer completion time among all qualified paths within all time windows, and Opt-MinT-CC returns the QR made on p at the end of the algorithm. Hence, we have $t \leq tw''^s + \frac{D}{min(B^{max}, B(p', tw''))}$, we then have $t \leq t''$, which contradicts our initial assumption that $t'' < t$.

Case (2): The edge adding loop stops after edge e'' is added. Since path p'' is the optimal data transfer path, we know that after we add edge e'' to G', v_s and v_d are connected by at least one path for the first time within current time window. As shown in Line 11 of Algorithm 3, Dijkstra's algorithm is used to identify the path with the least length from v_s to v_d through e'', suppose this path is p'. Since both p'' and p' share the same bottleneck edge e'', we know that $B(p'', tw'') = B(p', tw'')$. From our analysis in Case (1), we also have the conclusion that $t \leq t''$, which also contradicts our initial assumption that $t'' < t$.

In summary, each of the above two cases contradicts our initial assumption. Proof ends. □

5.2 Algorithm Design and Analysis of FBR-MinT-CC

5.2.1 Algorithm Design

Please refer to Algorithm 4 for the detailed algorithm design and pseudocode of FBR-MinT-CC. In the worst case, its complexity is $O\big(|td|^2 \cdot (|E| + |V| \cdot \log |V|)\big)$.

Algorithm 4 FBR-MinT-CC

GIVEN: $G(V, E)$
INPUT: MinT-CC BRR $(v_s, v_d, B^{max}, D, t^S, C^{max})$
OUTPUT: Estimated QRECT or $NULL$ if no QR can be identified

1: The same as Lines 1 – 3 of Algorithm 3;
2: **for** $i \leftarrow 0$ to $|ltw| - 1$ **do**
3: Run Dijkstra's algorithm to identify the path with the least length p' from v_s to v_d;
4: **if** $p' \neq NULL$ && $L(p') \leq \lfloor \frac{C^{max}}{D} \rfloor$ && $tw_i^s + \frac{D}{min(B^{max}, B(p', tw_i))} < t$ **then**
5: $t = tw_i^s + \frac{D}{min(B^{max}, B(p', tw_i))}$, $p = p'$ and $tw = tw_i$;
6: The same as Lines 15 – 18 of Algorithm 3.

5.2.2 Algorithm Illustration

We illustrate FBR-MinT-CC using G shown in Fig. 1 and the example MinT-CC BRR $(v_s, v_d, 12\,\text{Gb/s}, 54\,\text{Gb}, 0, 200)$.

Iterate through $ltw = \{[0, 6\,\text{s}], [0, 11\,\text{s}], [0, \infty), [6\,\text{s}, 11\,\text{s}], [6\,\text{s}, \infty), [11\,\text{s}, \infty)\}$. Within time windows $[0, 6\,\text{s}]$ and $[0, 11\,\text{s}]$, we know that the shortest path from v_s to v_d, $v_s - v_2 - v_d$, could not finish the data transfer of the example BRR. Within time window $[0, \infty)$, the shortest path from v_s to v_d is $v_s - v_2 - v_d$ with the available bandwidth of 3 Gb/s and length of 2, which is less than $\lfloor \frac{200}{54} \rfloor = 3$. The completion time of the data transfer on it is $\frac{54\,\text{Gb}}{3\,\text{Gb/s}} = 18\,\text{s}$. After the iteration of the rest of the time windows, we know we cannot find any QR with earlier completion time, and we have $t = 18\,\text{s}$, $p = v_s - v_2 - v_d$ and $tw = [0, \infty)$. The corresponding QR is $(v_s - v_2 - v_d, 3\,\text{Gb/s}, [0, 18\,\text{s}], 108)$, the estimated QRECT for the example BRR.

6 Performance Evaluation

One of the most widely used bandwidth reservation service in scientific area is the OSCARS of ESnet [28–32]. Currently more than 40 U.S. Department of Energy's research sites, including all national laboratory systems, and another 140 commercial and research institutions around the world are using the service provided by ESnet for their daily large-scale data movement. To make our performance evaluation as real and accurate as possible, topology of ESnet is drawn using the data gathered from ESnet [10, 33] to mimic the real ESnet scenario, and we then conduct intense simulations on the drawn topology.

We run 10 sets of simulations and for simulation i, $1 \leq i \leq 10$, 10 BRR batches containing $i \times 200$ BRRs are randomly generated. For MinC-TC BRR $(v_s, v_d, B^{max}, D, [t^S, t^E])$ and MinT-CC BRR $(v_s, v_d, B^{max}, D, t^S, C^{max})$, v_s and v_d are two randomly selected nodes from the node set $(v_s \neq v_d)$, B^{max} is a random integer within 1 and 10,000, $D \leq B^{max} \cdot (t^E - t^S)$, t^S is a random integer within the range [0, 19] while t^E is a random integer from the range $(t^S, 20]$, and C^{max} is the multiplication between D and a random integer within the range [1, 10]. All the proposed algorithms, Opt-MinC-TC, FBR-MinC-TC, Opt-MinT-CC and FBR-MinT-CC, are implemented to process the same batches of BRRs. Several performance metrics are collected after the BRR processing, and corresponding figures are drawn. To make our experiment results as accurate as possible, all figures in this section show both the average performance measurements of the performance metrics and the corresponding variances with the 95% confidence level across all the simulation sets.

6.1 Performance Analysis of Opt-MinC-TC and FBR-MinC-TC

After Opt-MinC-TC and FBR-MinC-TC finish the processing of all the BRRs in one batch, two performance metrics are collected: (1) BRR scheduling success ratio, defined as the percentage of BRRs that have been successfully scheduled within the BRR batch, and (2) average length of the data transfer paths of the successfully scheduled BRRs within the batch. The second metric is used to measure the data transfer cost of the scheduled BRRs based on our data transfer cost model proposed in Sect. 3.2.

After data analysis, we further plot the experimental results in Figs. 2 and 3. Suppose we use *LBRR* to denote one BRR batch, and s and s' to denote the set of MinC-TC BRRs within *LBRR* that can be successfully scheduled by Opt-MinC-TC and FBR-MinC-TC, respectively. Opt_MinC_TC_FBR in Figs. 2 and 3 denotes the ratio of the BRRs in one batch that can be successfully scheduled by both Opt-MinC-TC and FBR-MinC-TC, and the corresponding average length of the data transfer paths, respectively. We know that any $brr \in s'$ can also be successfully scheduled by Opt-MinC-TC and its cost computed by Opt-MinC-TC and FBR-MinC-TC is identical. The data analysis shows that Opt-MinC-TC can successfully schedule 10.69% more BRRs averagely in one BRR batch than FBR-MinC-TC, namely $\frac{|s|-|s'|}{|LBRR|} = 10.69\%$ in average as shown by Opt_MinC_TC_Extra in Fig. 2, and the average length of the data transfer paths of the scheduled BRRs by Opt-MinC-TC is 20.30% larger than that computed by FBR-MinC-TC (Fig. 3). From the above two parameters, we derive that $s' \subset s$ and for those BRRs that Opt-MinC-TC can successfully schedule while FBR-MinC-TC cannot, namely the BRRs in set $s - s'$, lengths of their data transfer paths are relatively longer than those of the BRRs both Opt-MinC-TC and FBR-MinC-TC can successfully schedule as shown by Opt_MinC_TC_Extra in Fig. 3.

In summary, Opt-MinC-TC successfully schedules much more BRRs in one batch with higher average length of the data transfer paths of the scheduled BRRs.

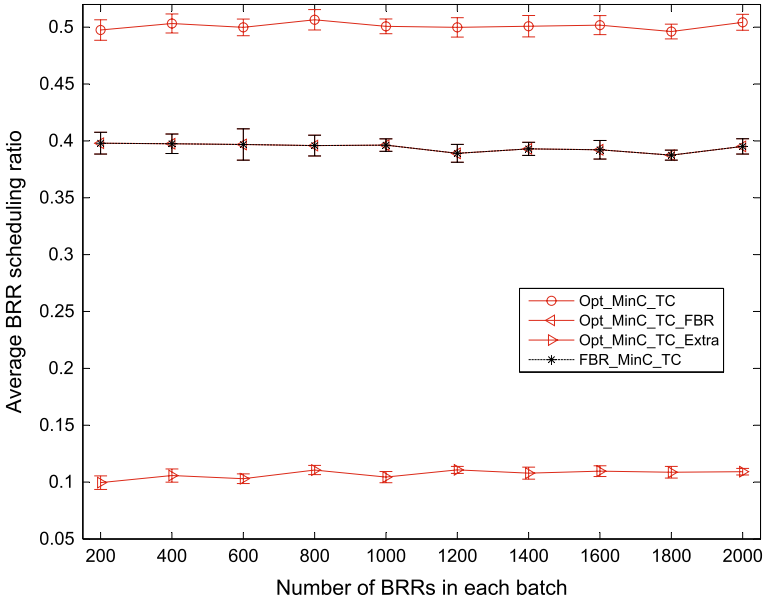

Fig. 2 Comparison of the BRR scheduling ratio by Opt-MinC-TC and FBR-MinC-TC

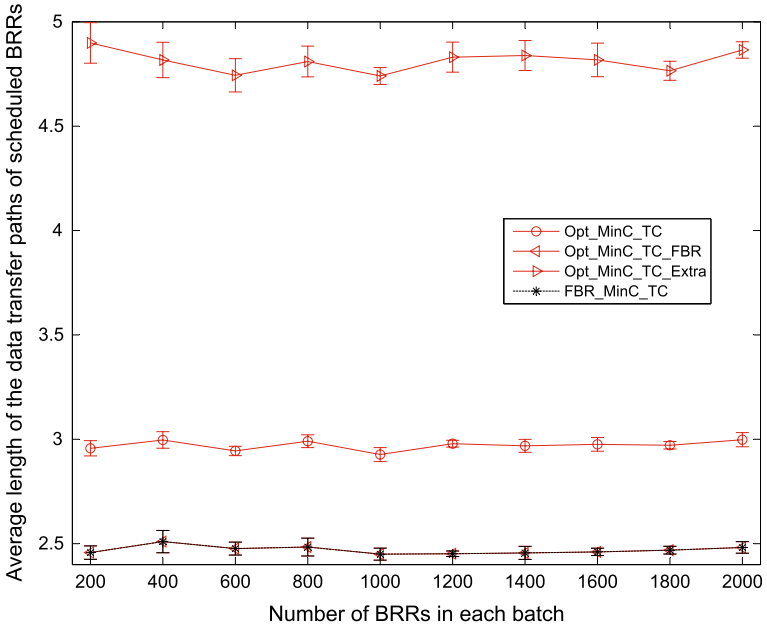

Fig. 3 Comparison of the average length of the data transfer paths of the scheduled BRRs by Opt-MinC-TC and FBR-MinC-TC

From our above result analysis, we know that Opt-MinC-TC has a much better overall scheduling performance than FBR-MinC-TC.

6.2 Performance Analysis of Opt-MinT-CC and FBR-MinT-CC

After Opt-MinT-CC and FBR-MinT-CC finish the processing of all the BRRs in one batch, similar performance metrics are collected: (1) BRR scheduling success ratio, and (2) average data transfer completion time of the successfully scheduled BRRs within the batch.

After data analysis, we further plot the experimental results in Figs. 4 and 5. Similarly, we also use set s and s' to denote the set of MinT-CC BRRs within a batch *LBRR* that can be successfully scheduled by Opt-MinT-CC and FBR-MinT-TC, respectively. The data analysis shows that the average BRR scheduling ratio computed by Opt-MinT-CC and FBR-MinT-CC is identical (Fig. 4), namely $s = s'$, and the average data transfer completion time of the successfully scheduled BRRs by FBR-MinT-CC is 7.89% higher than that computed by Opt-MinT-CC (Fig. 5).

From the experiment result, we know that if a MinT-CC BRR is schedulable, namely as long as the BRR can be successfully scheduled theoretically, both Opt-MinT-CC and FBR-MinT-CC can successfully schedule it. However, its data transfer completion time computed by Opt-MinT-CC and FBR-MinT-CC might be different, an example of which is shown in the illustrations of Opt-MinT-CC and FBR-MinT-CC.

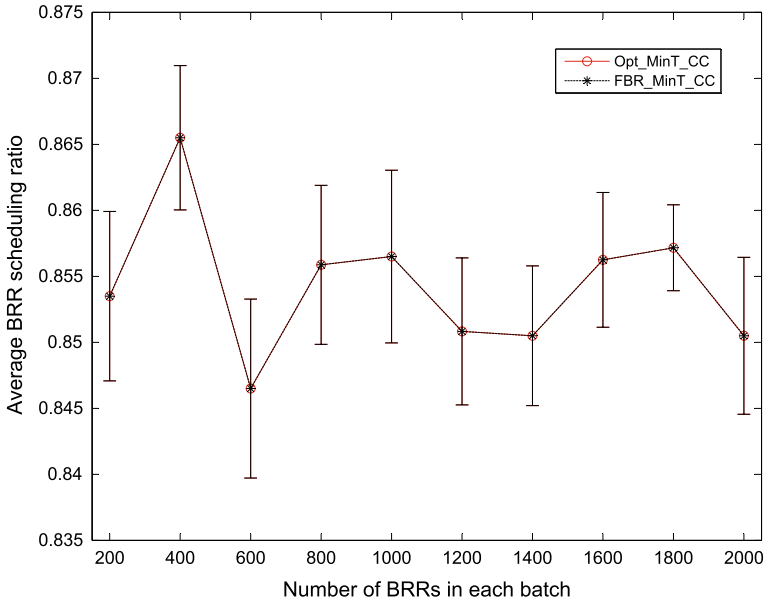

Fig. 4 Comparison of the BRR scheduling ratio by Opt-MinT-CC and FBR-MinT-CC

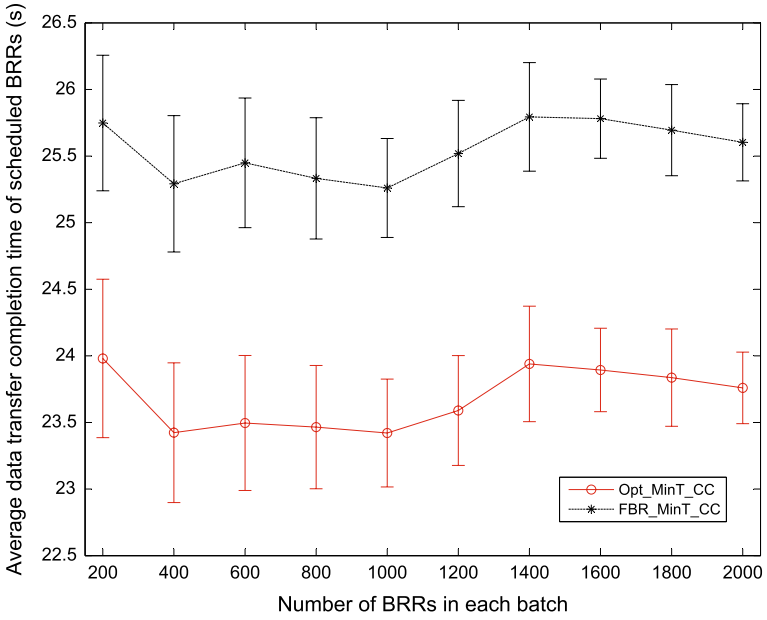

Fig. 5 Comparison of the average data transfer completion time by Opt-MinT-CC and FBR-MinT-CC

In summary, Opt-MinT-CC schedules the same amount of BRRs in one batch as FBR-MinT-CC, but with less average data transfer completion time of the successfully scheduled BRRs. From our above result analysis, we know that Opt-MinT-CC has a much better overall scheduling performance than FBR-MinT-CC.

7 Conclusion and Future Work

In this paper, we focused on the trade-off between cost and performance of data transfers using bandwidth reservation in dedicated networks. The most common data transfer performance parameter, data transfer completion time, was specifically studied. We considered the scheduling of two types of BRRs regarding such trade-off: (1) to achieve the minimum data transfer cost given the data transfer deadline, and (2) to achieve the earliest data transfer completion time given the maximum data transfer cost. We suppose the data transfer cost is proportional to the length of the data transfer path. We then proposed two bandwidth reservation algorithms with rigorous optimality proofs, i.e., Opt-MinC-TC and Opt-MinT-CC, to optimize the scheduling of each MinC-TC and MinT-CC BRR. We compared the proposed algorithms with two scheduling algorithms originating from one widely used scheduling algorithm in production networks, i.e., FBR-MinC-TC and FBR-MinT-CC. Extensive simulations were conducted on the topology of ESnet drawn using the real data collected from online, and different performance metrics were used to evaluation the scheduling performance of the proposed algorithms. The extensive simulation

results showed Opt-MinC-TC and Opt-MinT-CC have much better overall scheduling performance than FBR-MinC-TC and FBR-MinT-CC, respectively.

We plan to study the following issues in the near future: (1) BRRs with different priorities and how to break the bandwidth reservation of the BRRs with lower priorities to satisfy the requirements of those with higher priorities, (2) the bandwidth reservation service with guaranteed performance in Cloud environment.

References

1. Braden, R., ed., Zhang, L., Berson, S., Herzog, S., Jamin, S.: Resource reservation protocol (rsvp)—version 1 functional specification, RFC 2205 (1997). https://doi.org/10.17487/RFC2205
2. Degermark, M., Köhler, T., Pink, S., Schelén, O.: Advance reservations for predictive service in the internet. Multimed. Syst. **5**(3), 177–186 (1997)
3. Braden, R., Clark, D., Shenker, S.: Integrated services in the internet architecture: an overview. Technical Report (1994)
4. White, P.P.: Rsvp and integrated services in the internet: a tutorial. IEEE Commun. Mag. **35**(5), 100–106 (1997)
5. Zuo, L., Zhu, M.M., Wu, C.Q.: Bandwidth reservation strategies for scheduling maximization in dedicated networks. IEEE Trans. Netw. Serv. Manage. **15**(2), 544–554 (2018)
6. Wu, C.Q.: Bandwidth scheduling in overlay networks with linear capacity constraints. In: IEEE Conference on Computer Communications (INFOCOM 2017) (2017), pp. 1–9
7. Zuo, L., Zhu, M.M., Wu, C.Q., Zurawski, J.: Fault-tolerant bandwidth reservation strategies for data transfers in high-performance networks. Comput. Netw. **113**, 1–16 (2017)
8. Lin, Y., Wu, Q.: Complexity analysis and algorithm design for advance bandwidth scheduling in dedicated networks. IEEE/ACM Trans. Netw. **21**(1), 14–27 (2013)
9. Balman, M., Chaniotakisy, E., Shoshani, A., Sim, A.: A flexible reservation algorithm for advance network provisioning. In: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computer Network, Storage and Analysis, Washington, DC, USA, pp. 1–11 (2010)
10. Zuo, L., Zhu, M., Wu, C.: Fast and efficient bandwidth reservation algorithms for dynamic network provisioning. J. Netw. Syst. Manage. **23**(3), 420–444 (2015)
11. Sim, A., Balman, M., Williams, D., Shoshani, A., Natarajan, V.: Adaptive transfer adjustment in efficient bulk data transfer management for climate datasets. In: The 22nd IASTED International Conference on Parallel and Distributed Computing and System (PDCS) (2010)
12. Zuo, L., Zhu, M.M., Wu, C.Q.: Concurrent bandwidth reservation strategies for big data transfers in high-performance networks. IEEE Trans. Netw. Serv. Manage. **12**(2), 232–247 (2015)
13. ESnet Network. http://www.es.net/about/. Accessed 24 Jan 2018
14. Summerhill, R.: The new Internet2 network. In: 6th Global Lambda Integrated Facility (2006)
15. Zheng, X., Veeraraghavan, M., Rao, N., Wu, Q., Zhu, M.: Cheetah: circuit-switched high-speed end-to-end transport architecture testbed. IEEE Commun. Mag. **43**(8), 11–17 (2005)
16. Recio, J., Grasa, E., Figuerola, S., Junyent, G.: Evolution of the user controlled light path provisioning system. In: Proceedings of 2005 7th International Conference on Transparent Optical Network, vol. 1, pp. 263–266 (2005)
17. Japanese Gigabit Network II. http://www.jgn.nict.go.jp. Accessed 4 July 2018
18. Sahni, S., Rao, N., Ranka, S., Li, Y., Jung, E.-S., Kamath, N.: Bandwidth scheduling and path computation algorithms for connection-oriented networks. In: The Sixth International Conference on Networking, pp. 47–47 (2007)
19. GÉANT's Bandwidth on Demand. https://www.geant.org/. Accessed 4 July 2018
20. Zuo, L., Zhu, M.M., Wu, C.Q., Hou, A.: Intelligent bandwidth reservation for big data transfer in high-performance networks. In: IEEE International Conference on Communications (ICC 2018), Kansas City, MO (2018) (in press)
21. Zuo, L.: Bandwidth preemption for data transfer request with higher priority. In: 36th International Performance Computing and Communications Conference (IPCCC 2017), pp. 1–2 (2017)

22. Wang, Y., Wu, C.Q., Hou, A.: On periodic scheduling of bandwidth reservations with deadline constraint for big data transfer. In: 41st IEEE Conference on Local Computer Networks (LCN 2016), pp. 224–227 (2016)
23. Wang, Y., Wu, C.Q., Hou, A.: Periodic scheduling of deadline-constrained variable slot-bandwidth reservations for scientific collaboration. In: 26th IEEE International Conference on Computer Communication and Networks (ICCCN 2017), pp. 1–9 (2017)
24. Hou, A., Wu, C.Q., Fang, D., Wang, Y., Wang, M., Wang, T., Zhang, X.: Bandwidth scheduling with multiple variable node-disjoint paths in high-performance networks. In: 35th IEEE International Performance Computing and Communications Conference (IPCCC 2016), pp. 1–4 (2016)
25. Zuo, L., Zhu, M.M.: Improved scheduling algorithms for single-path multiple bandwidth reservation requests. In: The 10th IEEE International Conference on Big Data Science and Engineering (BigDataSE-16), pp. 1692–1699 (2016)
26. Zuo, L., Zhu, M.M., Wu, C.Q.: Concurrent bandwidth scheduling for big data transfer over a dedicated channel. Int. J. Commun. Netw. Distrib. Syst. **15**(2/3), 169–190 (2015)
27. Zuo, L., Zhu, M.: Toward flexible and fast routing strategies for dynamic network provisioning. In: 27th International Parallel and Distributed Processing Symposium PhD Forum, pp. 2222–2225 (2013)
28. Guok, C., Robertson, D., Thompson, M., Lee, J., Tierney, B., Johnston, W.: Intra and interdomain circuit provisioning using the Oscars reservation system. In: 3rd International Conference on Broadband Communication, Networks and Systems, pp. 1–8 (2006)
29. Charbonneau, N., Vokkarane, V.M., Guok, C., Monga, I.: Advance reservation frameworks in hybrid IP-WDM networks. IEEE Commun. Mag. **49**(5), 132–139 (2011)
30. Guok, C., Lee, J.R., Berket, K.: Improving the bulk data transfer experience. Int. J. Internet Protoc. Technol. **3**(1), 46–53 (2008)
31. Lehman, T., Yang, X., Ghani, N., Gu, F., Guok, C., Monga, I., Tierney, B.: Multilayer networks: an architecture framework. IEEE Commun. Mag. **49**(5), 122–130 (2011)
32. Monga, I., Guok, C., Johnston, W.E., Tierney, B.: Hybrid networks: lessons learned and future challenges based on esnet4 experience. IEEE Commun. Mag. **49**(5), 114–121 (2011)
33. ESnet Network Weathermap. https://my.es.net/. Accessed 24 Jan 2018

Liudong Zuo received the Ph.D. degree in computer science from Southern Illinois University Carbondale in 2015. He received the B.E. degree in computer science from University of Electronic Science and Technology of China in 2009. He is currently an assistant professor in Computer Science Department at California State University, Dominguez Hills. His research interests include computer networks, algorithm design, and big data.

Michelle M. Zhu is as an associate professor in Computer Science Department at Montclair State University. Prior to that, she was an associated professor in Computer Science Department of Southern Illinois University Carbondale. She completed her dissertation in the Computer Science and Mathematics division of Oak Ridge National Laboratories (ORNL). Her research areas focus on parallel and distributed computing and big data system.

Chia-Han Chang received the B.E.E. degree from National Taiwan University in 2014. He is currently pursuing his master degree in computer science at California State University, Dominguez Hills. His research interests include computer networks, algorithm design, and big data.