SC18 Dallas, TX hpc inspires.
Program November 11–16, 2018
Exhibits November 12–15, 2018
KAY BAILEY HUTCHISON CONVENTION CENTER DALLAS
The International Conference for High Performance Computing, Networking, Storage, and Analysis

Bandwidth Scheduling for Big Data Transfer with Deadline Constraint between Data Centers

Aiqin Hou[a] Chase Q. Wu[b] , Dingyi Fang[a], Liudong Zuo[a], Michelle M. Zhu[d], Xiaoyang Zhang[a], Ruimin Qiao[a], Xiaoyan Yin[a]

[a]School of Information Science and Technology, Northwest University, China
[b]New Jersey Institute of Technology, USA
[c]California State University Dominguez Hills, USA
[d]Montclair State University, USA

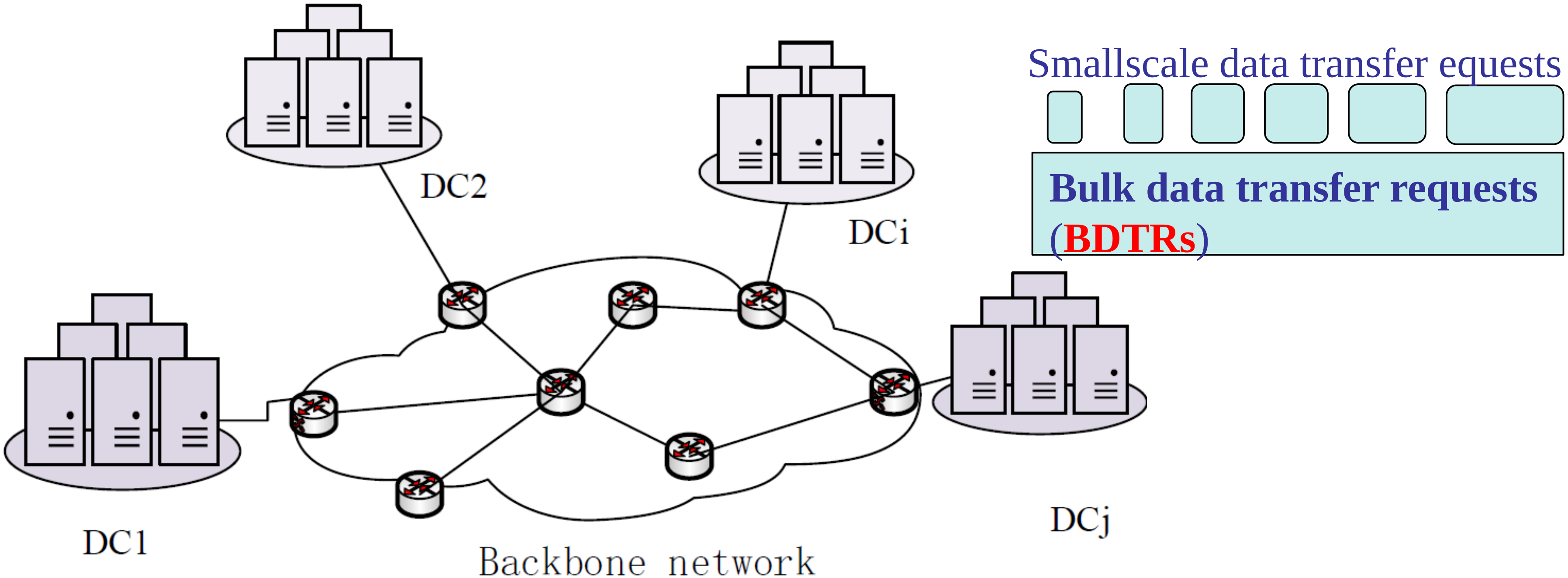
Outline

- Introduction
- Problem Formulation
- Algorithm Design
- Performance Evaluation
- Conclusion

Outline

- **Introduction**
- Problem Formulation
- Algorithm Design
- Performance Evaluation
- Conclusion

Big Data *Transfer Between Data Centers*



Different types of inter-DC data transfer: **BDTR** account for a major portion of traffic.

There exist many state of the art research work about bulk data transfer. However, most existing solutions for BDTRs are tailored for private cloud services, hence limiting their generalization and scope of application.

High-performance networks (**HPN**s)

Typical characteristics of HPN include:

- **Links with high capacity – up to 100 Gbps**
- **Capable of bandwidth reservation**

Bandwidth Scheduling (BS) - Architecture

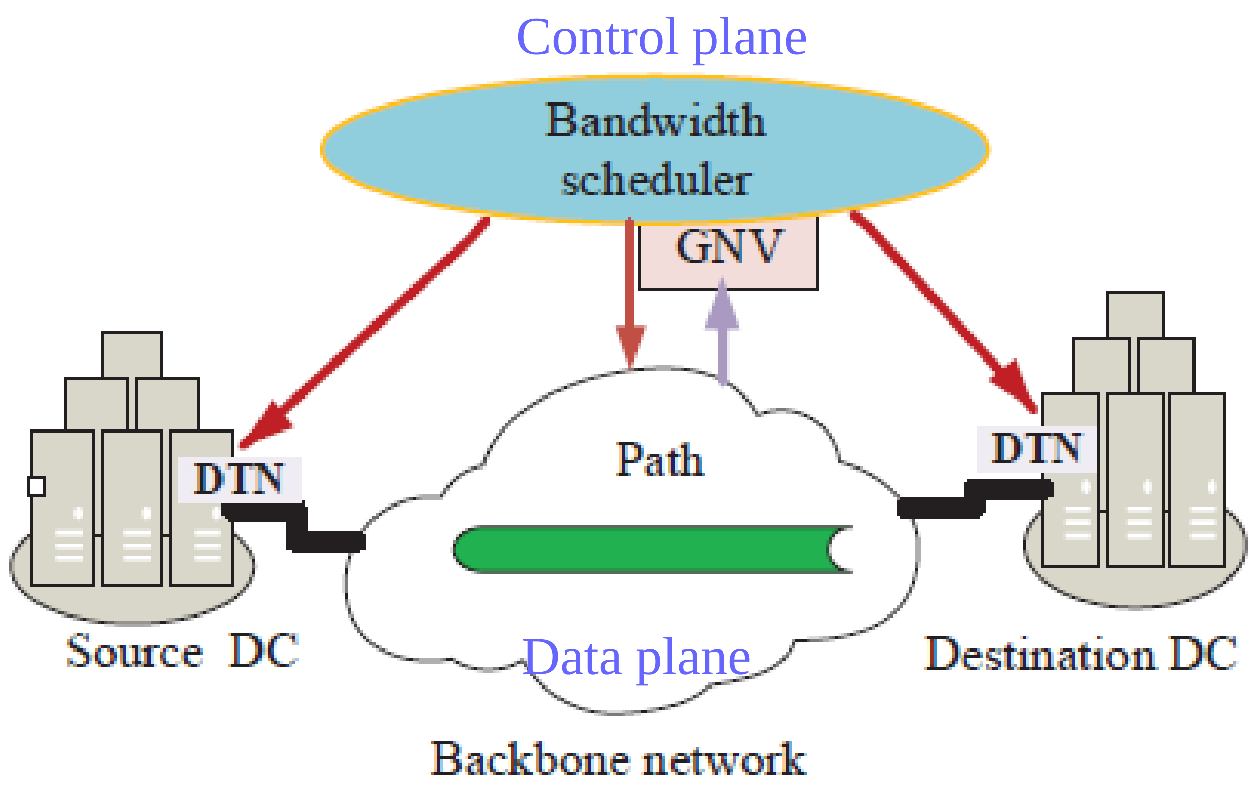


Fig. 1: An example HPN with bandwidth reservation service.

Bandwidth Scheduling (BS)-Our work

We investigate a bandwidth scheduling problem for **two types of BDTRs with fixed or variable bandwidth**. Our works include:

- ✓ Construct rigorous cost models to define a new performance metric of ***user satisfaction degree(USD)***;
- ✓ Formulate a generic problem **BS-MRVT** and prove its *NP-Completeness* and *nonapproximable*.
- ✓ Design a heuristic algorithm **FMS-MRVT** for the BS-MRVT.

Outline

- Introduction
- **Problem Formulation**
- Algorithm Design
- Performance Evaluation
- Conclusion

> - ***Mathematic Models***
> - ***Problem Definition***
> - ***Complexity analysis***

Mathematic Model

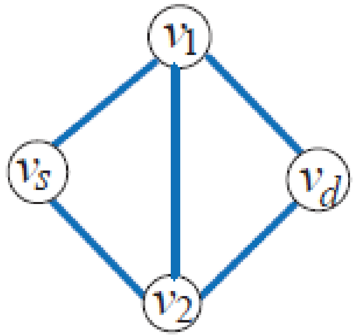


Fig. 2: An example network topology.

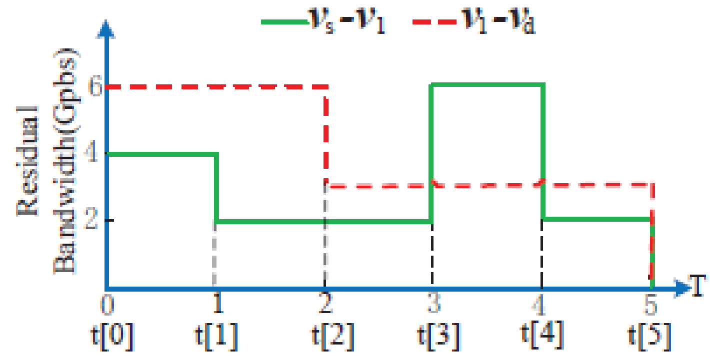


Fig. 3: Aggregated time bandwidth list of two links.

- An ATB (aggregated time bandwidth) of all links can be denoted as:

$$(t[0], t[1], b_0[0], b_1[0], ..., b_{|E|-1}[0]), \ \ldots (t[T-1], t[T], b_0[T-1], b_1[T-1], ..., b_{|E|-1}[T-1]),$$

where ***T*** is the total number of new time-slots after the aggregation of TB lists of all |***E***| links.

Mathematic Model– BDTRs

BDTR : $\boldsymbol{r}_{\mathbf{i}}$ $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, b_i^f)$

source node

destination node

deadline

data size

the maximum local area network (LAN) bandwidth

a boolean variable

the default earliest data transfer start time is 0.

- when $\boldsymbol{b_i^f}$ = true, r_{i} is **FBRR** (Fixed Bandwidth Bulk data Request);
- Otherwise, it's **VBRR** (Variable Bandwidth Bulk data Request).

Problem Definition -USD(1)

In general, given multiple concurrent user requests, the **scheduling success ratio (SSR),** serves as a good indicator for scheduling performance.

- We **define a new performance metric** of *User Satisfaction Degree* (USD) to quantify the transfer performance of each individual user request.

The USD of each $\boldsymbol{r}_{\mathbf{i}}$, denoted by usd_i, is defined as:

$$usd_i = a_i \cdot (t_i^d/(t_i^E + t_i^d)) \qquad \textbf{(1)}$$

when $a_i = 0$ ($\boldsymbol{r}_{\mathbf{i}}$ is not accommodated), $\boldsymbol{usd}_{\mathbf{i}} = 0$;

when $a_i = 1$, usd_i is maximized if t_i^E is minimized.

$$t_i^E \in (0, t_i^d] \qquad usd_i \in [0.5, 1)$$

Problem Definition - USD (2)

Given a set of BDTRs to be scheduled, we calculate the sum of the USD of all BDTRs as:

$$usd = \sum_{r_i \in BDTR} usd_i = \sum_{r_i \in BDTR} \left(a_i \cdot \frac{t_i^d}{t_i^E + t_i^d}\right) \quad \textbf{(2)}$$

subject to

$$\sum_{r_i \in BDTR} a_i \cdot b_i^l(t) \leq C^l, \forall l \in E, \forall t \in [0, T]. \quad \textbf{(3)}$$

where $\boldsymbol{C_l}$ represents the bandwidth capacity of link $\boldsymbol{l}$ and $\boldsymbol{b^l_i(t)}$ denotes the reserved bandwidth for $\boldsymbol{r_i}$ on link $\boldsymbol{l}$ within time slot $\boldsymbol{t}$.

Problem Definition - **BS-MRVT**

We formally define **BS-MRVT** (Bandwidth Scheduling for Multiple Reservations of Various Types) as follows:

BS-MRVT Definition: Given a backbone network ***G*(*V*,*E*)** with an ***ATB*** list for all links and multiple **BDTRs (v_i^s, v_i^d, t_i^d, δ_i, b_i^{max}, b_i^f)** of two types, either FBBR or VBBR, we wish to maximize the ***SSR*** and the total ***USD*** of all BDTRs defined in ***Eq*. 2** under the constraint of ***Eq.*** 3.

Complexity Analysis

BS-MRVT are both NP-complete and nonapproximable.

A known NP-hard problem

SSUSF (**single-sink un-splittable flow**) [1] problem :
Given ***G (V, E)*** with a bandwidth capacity of each link , and a batch of demands$\{D_1, D_2, \dots, D_k\}$, each D_i denoted as $(s_i, t, w_i=1, d_i, C_i)$,
The goal is to find a schedule that **maximizes the number of demands routed successfully under the link bandwidth capacity constraints**.

[1] F. B. Shepherd and A. R. Vetta, "The inapproximability of maximum single-sink unsplittable, priority and confluent flow problems," Theory of Computing, vol. 13, no. 20, pp. 1–25, 2017.

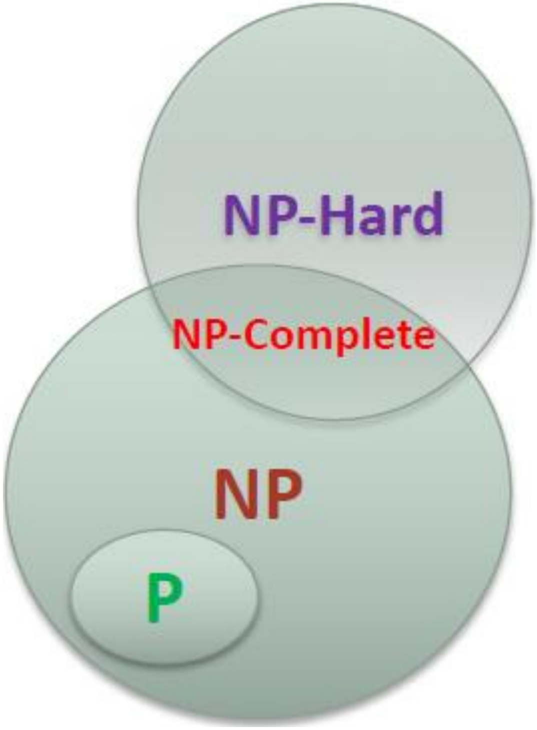
*Complexity Analysis --**NP-complete** (1)*

Theorem 1. *BS-MRVT* is NP-complete.

Proof*.** The decision version of BS-MRVT is as follows: ***G(V,E) with an ATB list for all links, and multiple **BDTRs** $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, b_i^f)$, does there exist a scheduling strategy such that the ***SSR* >=*m*** and ***USD* >= *n***?

■ BS-MRVT is NP .

Given the scheduling options of all BDTRs, we can calculate the ***SSR*** and ***USD***, and verify the correctness of the answer in polynomial time. Hence, **BS-MRVT is NP**.



Complexity Analysis --**NP-complete** (2)

■ **BS-MRVT is NP-hard.**

A problem is NP-hard if a special case of this problem with a particular input is equivalent to a known NP-hard problem [2].

We **construct a special case of BS-MRVT with a particular structure**, as shown in **Fig. 4**:

- A **special request** r_i in the form of **$(v_i^s,\ v^d,\ 1,\ \delta_i,\ \delta_i,\ true)$,**
- Further consider **a specific network topology** where the bandwidth could only be reserved on one single path.

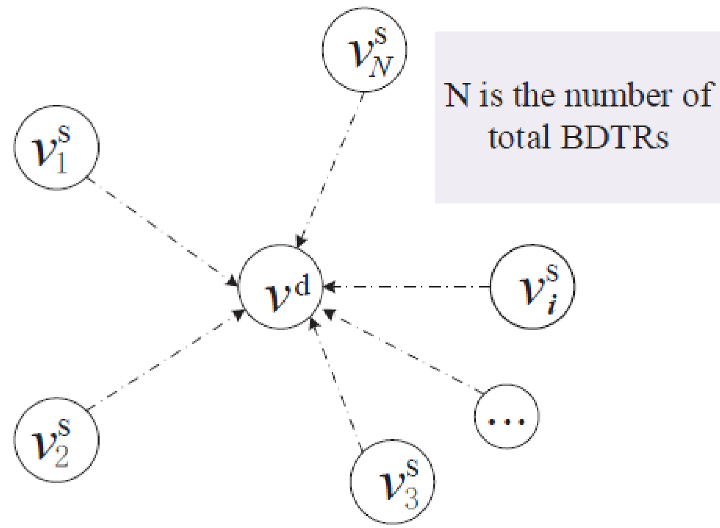


Fig. 4: An instance of BS-MRVT with a particular structure

Therefore maximize :

$$usd = \sum_{r_i \in BDTR} usd_i = \sum_{r_i \in BDTR} \left(a_i \cdot \frac{t_i^d}{t_i^E + t_i^d}\right) \Longrightarrow \sum_{r_i \in BDTR} a_i$$

[2] R. L. R. T. H. Cormen, C. E. Leiserson and C. Stein, Introduction to Algorithms, 3rd ed. MIT Press, Cambridge, MA, USA, 2009.

*Complexity Analysis --**NP-complete** (3)*

We now show that any instance of the known NP-hard problem SSUSF(unit-profit case) [1] with the objective to maximize cardinality can be transformed into **the above instance of BS-MRVT** in polynomial time.

BS-MRVT $\boldsymbol{r}_{\mathbf{i}}$ $(v_i^s, v^d, 1, \delta_i, \delta_i, true)$ |BDTRs|=k
(special instance)

SSUSF[1] $\boldsymbol{D}_{\mathbf{i}}$ $(s_i, t, 1, d_i, C_i)$ Demands$\{D_1, D_2, \ldots D_k\}$

Obviously, if we have a solution to the instance of SSUSF, we have a solution to the instance of BS-MRVT with a particular structure, and vice versa.

Since a special case of BS-MRVT with a particular structure is NP-complete, so is the original BS-MRVT problem.

*Complexity Analysis–**Nonapproximable**(1)*

Theorem 2. ***BS-MRVT is Nonapproximable***.

Proof . From Theorem 1.3 in [1], we know that we cannot approximate SSUSF with a factor of $O(|E|^{1/2-\epsilon})$ for any $\epsilon > 0$.
Assume that there exists an approximate algorithm with an approximation ratio of $O(|E|^{1/2-\epsilon})$ for a certain $\epsilon > 0$ for BS-MRVT.
We show that this assumption implies a polynomial time optimal solution to SSUSF [1].

SSUSF (unit-profit case): $D_i\ (s_i, t, 1, d_i, C_i)$,

The objective of SSUSF is to maximize the number of demands routed successfully under the link bandwidth capacity constraints.

*Complexity Analysis–**Nonapproximable**(2)*

We then construct a corresponding instance of BS-MRVT in polynomial time.

Firstly, consider a special BDTR r_i by setting the parameter set in **(v_i^s, v^d, *1*, δ_i, δ_i, *true*)** to (s_i, t, 1, d_i, C_i).

Secondly, consider a specific network topology for BS-MRVT as shown in Fig. 4,

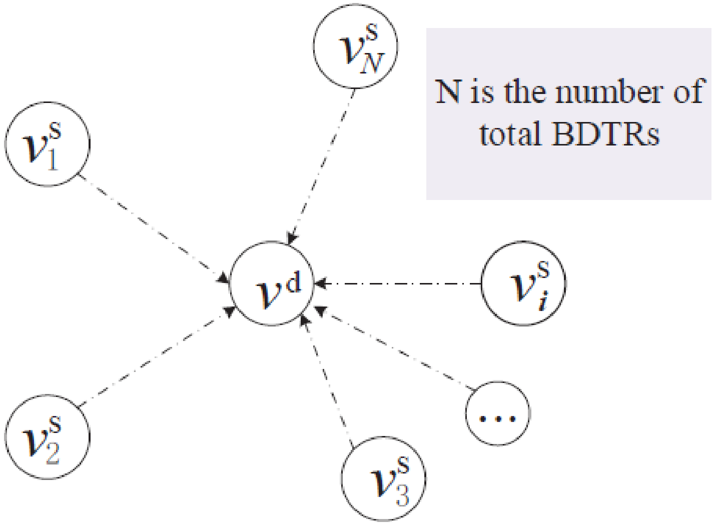


Fig. 4: An instance of BS-MRVT with a particular structure

*Complexity Analysis–**Nonapproximable**(3)*

Hence, maximizing the total USD of all BDTRs is equivalent to maximizing the number of BDTRs that are successfully scheduled. Obviously, **it equivalent to the SSUSF** (unit-profit case) **problem that maximizing the number of demands routed successfully**.

We apply the assumed approximate algorithm to the instance of BS-MRVT as described above. Obviously, the assumed approximate algorithm for BS-MRVT finds an optimal solution to SSUSF (unit-profit case) whenever one exists. This conflicts with the NP-completeness of SSUSF. □

Outline

- Introduction
- Problem Formulation
- Algorithm Design
- Performance Evaluation
- Conclusion

> A. Design of a Heuristic Algorithm **FMS-MRVT** for BS-MRVT
> B. Illustration of FMS-MRVT

A. *Algorithm Design of* FMS-MRVT

FMS-MRVT (**F**lexible **M**ultiple **S**cheduling for **MRVT**). The pseudocode is provided in **Algorithm 1**.

Algorithm 1 FMS-MRVT

Input: An HPN graph $G(V, E)$ with an ATB list of all links, multiple BDTRs $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, b_i^f)$

Output: the total user satisfaction degree usd

Initialize variable $usd = 0$;

Sort all BDTRs by their deadlines in an ascending order. For BDTRs with the same deadline, further sort them by their data sizes in an ascending order, and for BDTRs with the same data size, FBBRs are placed ahead of VBBRs;

for each $r_i : (v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, b_i^f)$ in the set of BDTRs **do**

  if $b_i^f == true$ **then**

    Call Algorithm 2;

  else

    Call Algorithm 3;

  $usd_i = a_i \cdot \frac{t_i^d}{t_i^E + t_i^d}$;

  $usd = usd + usd_i$;

return usd.

In the worst case, its time complexity is $O(|BDTR|T^2 \ (|E| + |V| \ \log(|V|))$

Algorithm Design for FBBR

Lines 3-19: We consider each time slot **q** within the range **[0, k].**

- Lines7-9, consider each time slots [p, q] in the reverse order, and calculate **VPFB** paths within the time slot range [p, q] (Line 9).
- Line 12 calculate the **FPFB** path within the time slots range [p, q],
- Line 13-19 chose the larger bandwidth of the FPFB and the VPFB determine if the data transfer could be completed. If so, set a_i=1 and calculate t_i^E; Otherwise, q++;

Algorithm 2 FBBR Scheduling

Input: an HPN graph $G(V, E)$ with an ATB list of all links, an FBBR r_i $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, true)$

Output: a_i to denote whether or not r_i could be successfully scheduled, and ECT t_i^E of r_i if $a_i = 1$

1: Initialize variables $a_i = 0$ and $t_i^E = \infty$;
2: Identify the time slot k such that $t[k] \leq t_i^d \leq t[k+1]$;
3: **for** $0 \leq q \leq k$ **do**
4:   Initialize variable $b_q = +\infty$;
5:   **for** each $l \in E$ **do**
6:     $b_l = min(b_i^{max}, C^l)$;
7:   **for** $q \geq p \geq 0$ **do**
8:     Use a modified Dijkstra's algorithm to compute the path with the maximum bandwidth b from v_i^s to v_i^d within time slot p;
9:     $b_q = min(b_q, b)$;
10:     **for** each $l \in E$ **do**
11:       $b_l = min(b_l, b_l[p])$;
12:     Use a modified Dijkstra's algorithm to compute the path with the maximum bandwidth b' from v_i^s to v_i^d within time slots $[p, q]$;
13:     $b_{[p,q]} = max(b_q, b')$;
14:     **if** $b_{[p,q]} \cdot \left(min(t[q+1], t_i^d) - t[p]\right) \geq \delta_i$ and $t[p] + \frac{\delta_i}{b_{[p,q]}} < t_i^E$ **then**
15:       $a_i = 1$;
16:       $t_i^E = t[p] + \frac{\delta_i}{b_{[p,q]}}$;
17:   **if** $a_i == 1$ **then**
18:     Update the residual bandwidths on the corresponding path;
19:     Break;
20: **return** a_i and t_i^E.

Algorithm Design for VBBR

Line 5 - 16:
consider each time slot within the range [0, k]. If the remaining data of $\boldsymbol{r_i}$ can be successfully transferred within time slot q, then we set $\boldsymbol{a_i}$ to 1 (Line 10) and calculate the data transfer completion time (Line 11); Otherwise, continue to next time slot.

Algorithm 3 VBBR Scheduling

Input: an HPN graph $G(V, E)$ with an ATB list of all links, a VBBR r_i $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, false)$

Output: a_i to denote whether or not r_i could be successfully scheduled, and ECT t_i^E of r_i if $a_i = 1$

1: Initialize variable $a_i = 0$ and $q = 0$;
2: Identify the time slot k such that $t[k] \leq t_i^d \leq t[k+1]$;
3: **for** each $l \in E$ **do**
4:     $b_l = min(b_i^{max}, C^l)$;
5: **while** $\delta_i > 0$ && $q \leq k$ **do**
6:     **for** each $l \in E$ **do**
7:         $b_l = min(b_l, b_l[q])$;
8:     Use a modified Dijkstra algorithm to compute the path $p_i[q]$ with the maximum bandwidth $b_i[q]$ from v_i^s to v_i^d within time slot q;
9:     **if** $\delta_i \leq b_i[q] \cdot \left(min(t_i[q+1], t_i^d) - t_i[q]\right)$ **then**
10:         $a_i = 1$;
11:         $t_i^E = t_i[q] + \frac{\delta_i}{b_i[q]}$;
12:         Update the residual bandwidths of the links on paths $p_i[0], p_i[1], \ldots, p_i[q]$;
13:         $Break$;
14:     **else**
15:         $\delta_i - = b_i[q] \cdot (t_i[q+1] - t_i[q])$;
16:     $q + +$;
17: **return** a_i and t_i^E.

B. *Illustration of* FMS-MRVT(1)

- The example network topology in **Fig. 2**
- The available bandwidths of the network links in time slots [0, 5] are shown in **Table I**:

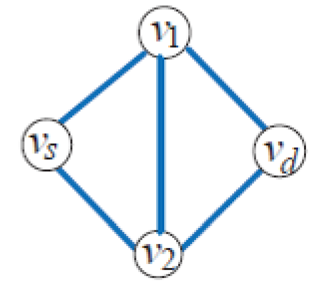


Fig. 2: An example network topology.

TABLE I: Available bandwidths of the links in Fig. 2.

Bandwidths (Gb/s) \ Time slots / Links	0	1	2	3	4	5
$v_s - v_1$	4	2	2	6	2	9
$v_1 - v_d$	6	6	3	3	3	5
$v_1 - v_2$	3	4	5	6	7	10
$v_s - v_2$	1	16	18	13	11	10
$v_2 - v_d$	6	14	17	9	10	18

- A set of **BDTRs** :
  - $r_1(FBBR) : (v_s, v_d, 4s, 10Gb, 15Gb/s, true)$
  - $r_2(FBBR) : (v_s, v_d, 4s, 15Gb, 17Gb/s, true)$
  - $r_3(VBBR) : (v_s, v_d, 5s, 20Gb, 12Gb/s, false)$

D. Illustration of FMS-MRVT(2)

Step 1: call **Algorithm 2** to schedule r_1, The computed FPFB path is v_s-v_2-v_d , the maximum fixed bandwidth is 14*Gb/s* within time interval [1*s*, 2*s*]. and calculate

$$usd_1 = \frac{6}{6+1.714} = 0.778$$

Step 2: call **Algorithm 2** to schedule r_2, and calculate

$$usd_2 = \frac{4}{4+2.882} = 0.5812.$$

Step 3: call **Algorithm 3** to schedule r_3, and calculate

$$usd_3 = \frac{5}{5+4.3} = 0.538.$$

Step 4: All of these three BDTRs can be successfully scheduled, and the overall USD of these three BDTRs is calculated as

$$usd = 0.778 + 1.388 + 0.538 = 1.897$$

$$SSR = 100\%.$$

Outline

- Introduction
- Problem Formulation
- Algorithm Design
- Performance Evaluation
- Conclusion

> - ***A.Simulation Setup***
> - ***B.Performance Comparison in ESnet5***
> - ***C. Performance Evaluation in Randomly Networks***
> - ***D. Performance Evaluation with Different Reservation Loads in Different Networks***

*A. Simulation **Setup***

We set the total time slots to span across 20 time units, and the start time ***t*[0] = 0**.
The link bandwidths follow a normal distribution:

$$b = b^{max} \cdot e^{-\frac{1}{2}(x)^2}$$

100Gb/s

a random variable within the range of (0, 1].

In each run of the simulation, we randomly generate 100-1500 BDTRs $(v_i^s, v_i^d, t_i^d, \delta_i, b_i^{max}, b_i^f)$

true or false

two randomly selected nodes

1 to 20

1Gbps to 20Gbps

no larger than $b_i^{max} \cdot t_i^d$

B. *Performance Comparison in* ***ESnet5***

To mimic the real ESnet scenario, we perform our simulations on the ESnet topology in Fig. 5

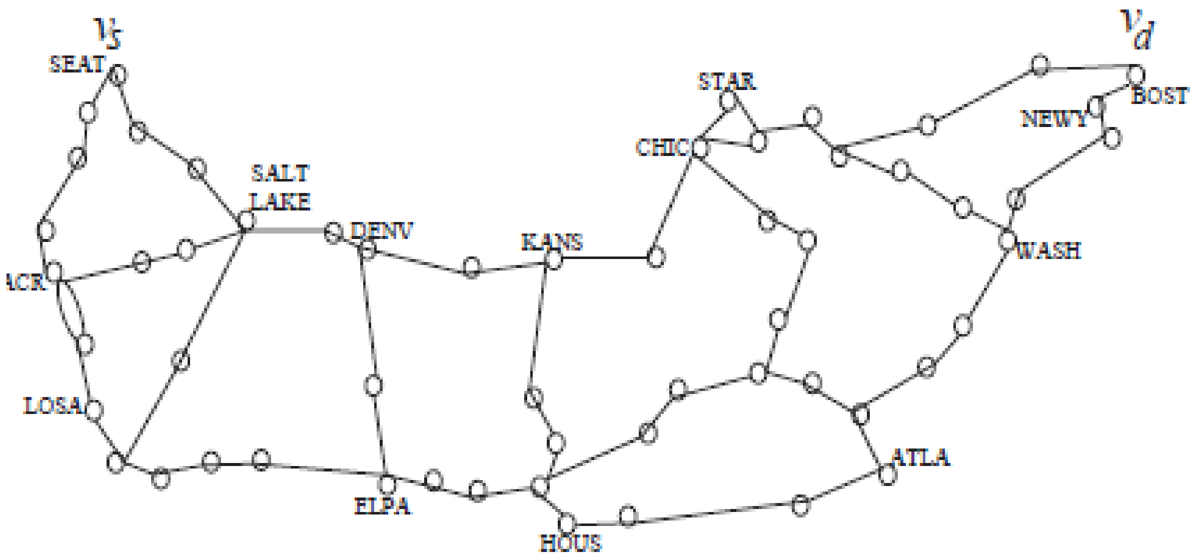


Fig. 5: The topology of ESnet [3].

[3] ESnet. https://www.es.net.

*B. Performance Comparison in **ESnet5***

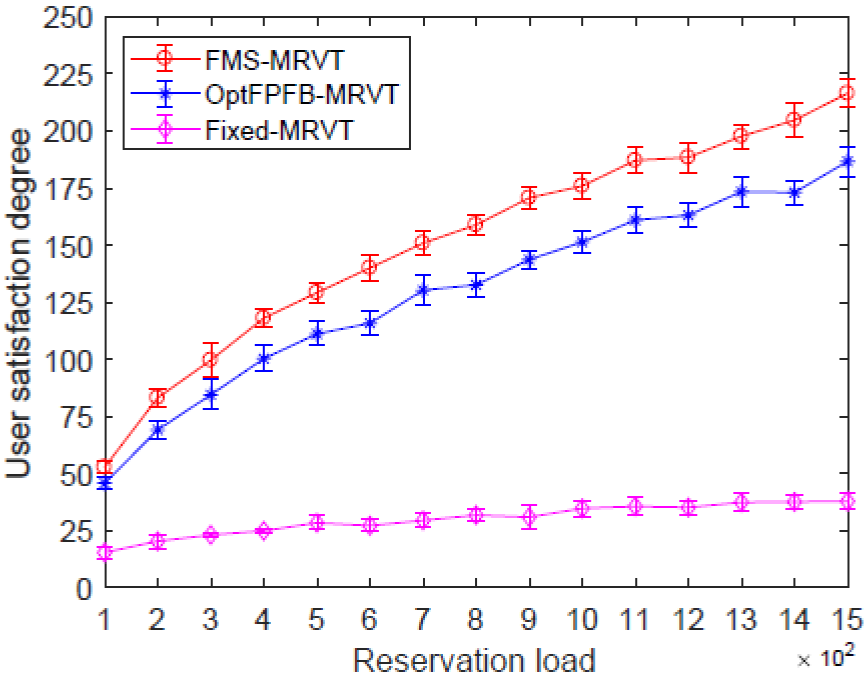


Fig. 6: USD comparison of three algorithms in ESnet:

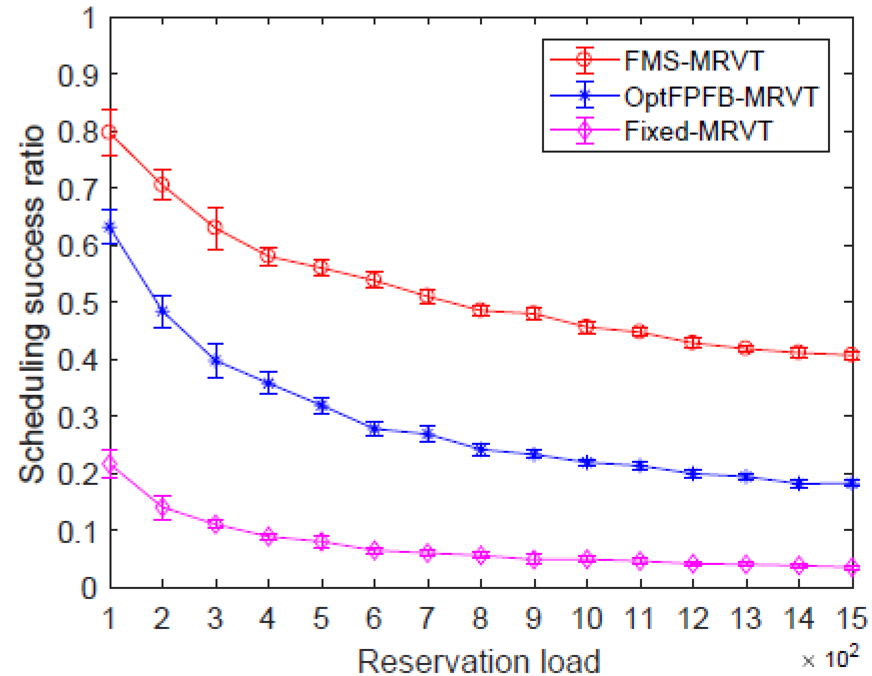


Fig. 7: SSR comparison of three algorithms in ESnet5.

We observe that FMS-MRVT outperforms OptFPFB-MRVT and Fixed-MRVT by 18-22% and 15-20% in terms of **USD** (As shown in Fig.6).,

We also observe that FMS-MRVT outperforms OptFPFB-MRVT **and** Fixed-MRVT by 50% and 3-5 times in terms of **SSR** (Fig.7).

C. Performance Evaluation in **Randomly Networks**

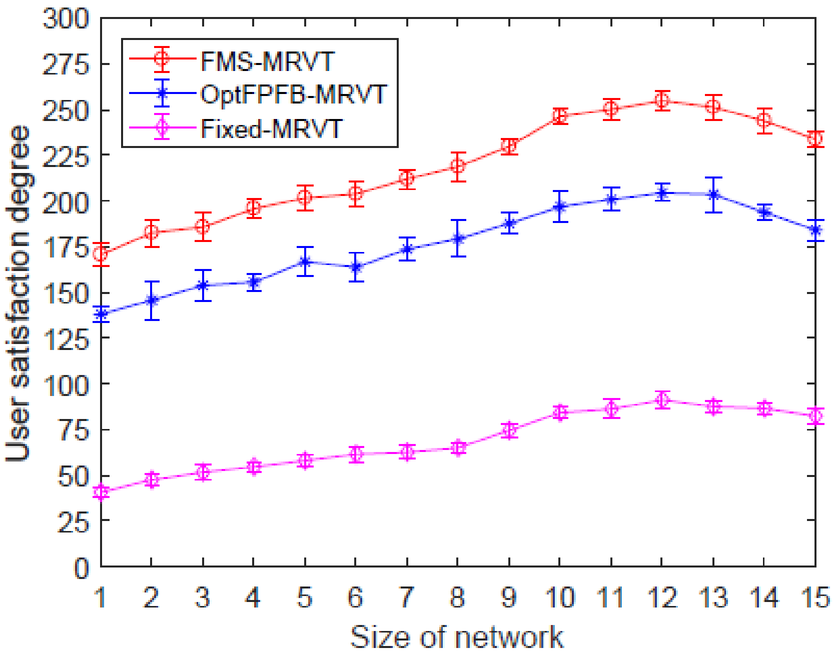


Fig. 8: USD comparison of three algorithms with 500 BDTRs in different random networks.

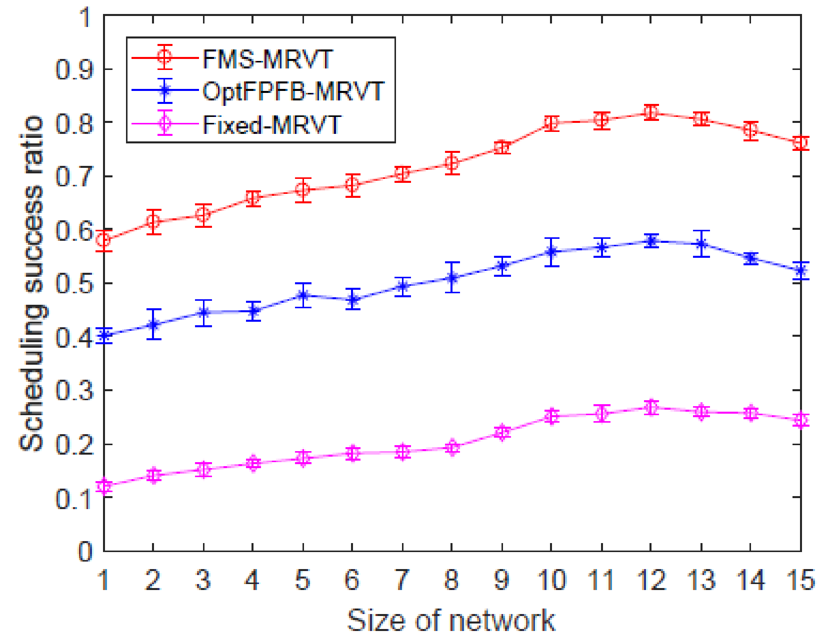


Fig. 9: SSR comparison with 500 BDTRs in different random networks.

TABLE II: Index of 15 large-scale networks.

Index of network	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of nodes	40	50	60	70	80	90	100	120	150	200	230	260	290	320	350
Number of links	80	100	120	140	160	180	200	240	300	400	450	500	520	540	560

Fig. 8 and 9 show that FMS-MRVT outperforms OptFPFB-MRVT and Fixed-MRVT by 23-26% and 17-24% in terms of USD, and 50% and 3 times in terms of SSR, respectively.

➢D. Performance Evaluation with **Different Loads** in **Different Networks**

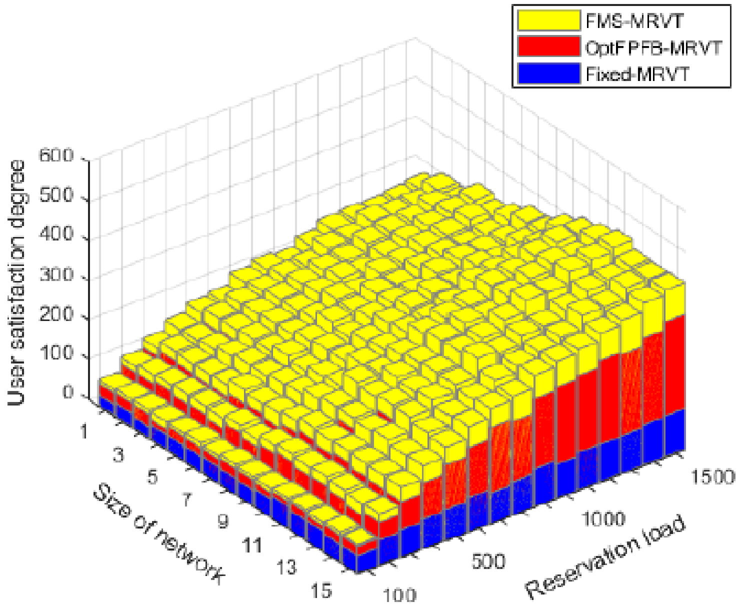


Fig. 10: USD comparison with variable loads in different networks.

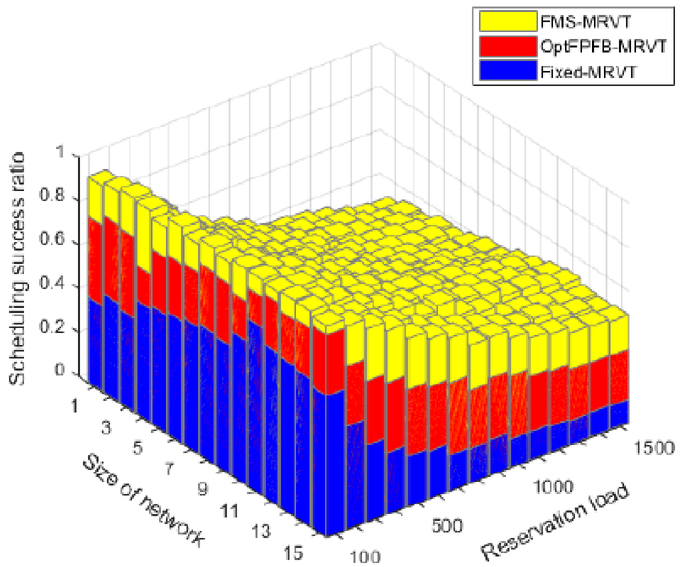


Fig. 11: SSR comparison with variable loads in different networks.

We observe that FMS-MRVT achieves consistently better performance than OptFPFB-MRVT and Fixed-MRVT in terms of both USD and SSR. (Fig. 10 and Fig. 11)

Outline

- Introduction
- Problem Formulation
- Algorithm Design
- Performance Evaluation
- Conclusion

Conclusion

- ✓ Formulate an advance bandwidth **problem** **BS-MRVT** with the objective to maximize BDTRs scheduling success ratio while minimizing the data transfer completion time of each request;
- ✓ Prove the **NP-completeness** and **Nonapproximable** of **BS-MRVT**;
- ✓ Propose **heuristics algorithms** **FMS-MRVT**, Extensive results show that the proposed algorithm achieve significantly **outperform than two other algorithms**.

Thank you

Q & A ?