Co-Scheduling of Advance and Immediate Bandwidth Reservations for Inter-Data Center Transfer

Aiqin Hou[a] Chase Q. Wu[b] , Liudong Zuo[c], Dawei Quan[a], Yangyang Li[a], Michelle M. Zhu[d], Qiang Duan[e], Dingyi Fang[a]

[a]Northwest University, China
[b]New Jersey Institute of Technology, USA
[c]California State University Dominguez Hills, USA
[d]Montclair State University, USA
[e]Pennsylvania State University, USA

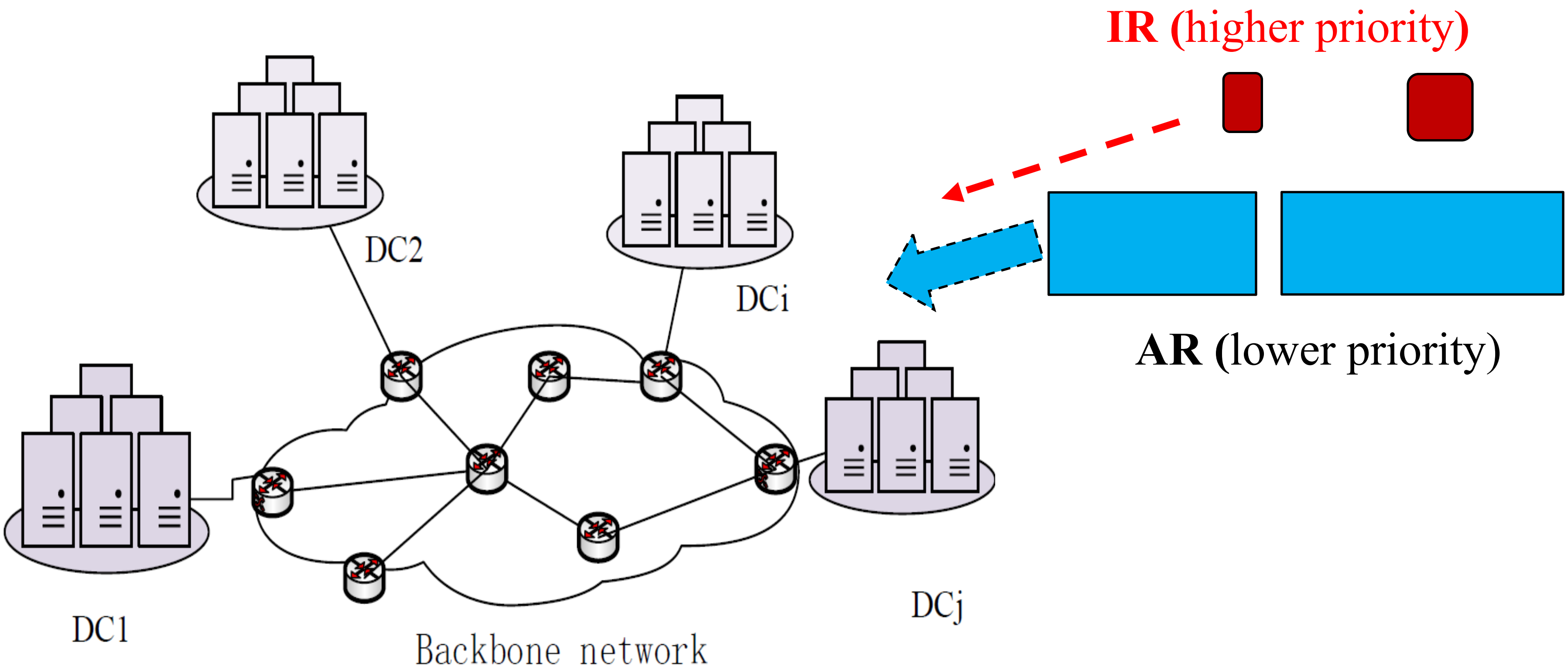
Outline

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

Outline

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

Big Data *Transfer Between Data Centers*



- **Bandwidth-preemption**: when an **IR** *with a higher priority* arrives, if the network is heavily loaded, a **connection preemption** may occur (***i.e.***, the bandwidth scheduler preempts *some existing* **AR** *with a lower priority* to free up bandwidths to accommodate the IR).

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

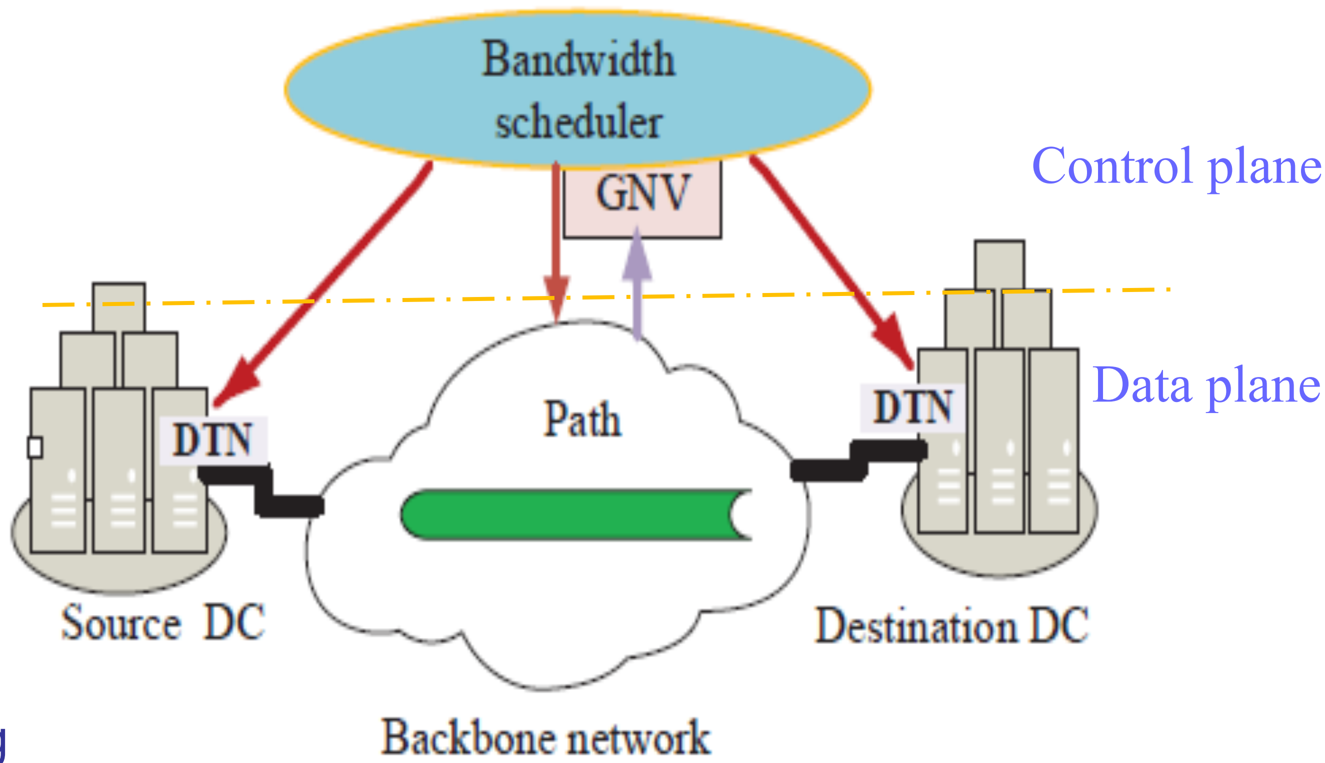
Esnet OSCARS and **Internet2 ION,** known as High-performance Networks (**HPNs**), which offer IP-based **MPLS** tunnels for various bandwidth reservation services.

Nowadays, many modern WAN backbones that connect geographically distributed DCs, can employ **SDN** technologies to create **HPNs**, which provide bandwidth reservation for big data transfer.

SDN-based Bandwidth Scheduling (BS) Architecture

A control plane with Global Network View (**GNV**) provides real-time network status information to the bandwidth Scheduler.

And the **bandwidth scheduler** is responsible for reserving or releasing bandwidths for user requests.



Collaborative Scheduling -- Our work

We investigate a co-scheduling problem **BS-ARIR** for two types of requests: **AR** and **IR** with different priorities. Our work includes:

- Construct two types of user request models: AR and IR;
- Define a performance metric of overall user *satisfaction* (***SAT***) to quantify users' Quality of Experience (QoE);
- Formulate a generic problem **BS-ARIR** and prove its ***NP-completeness***.
- Design heuristic scheduling algorithms ***Min-R-AR*** for periodic **ARs** and ***Max-S-ARIR*** for collaborative scheduling of **ARs and IRs**.

Outline

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

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

Mathematical Model - HPN

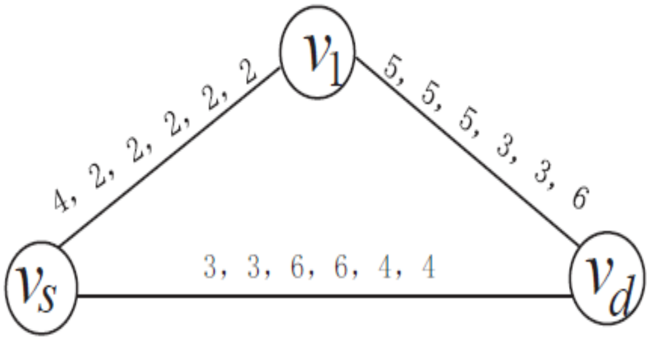


Fig. 1: An HPN example of a simple topology.

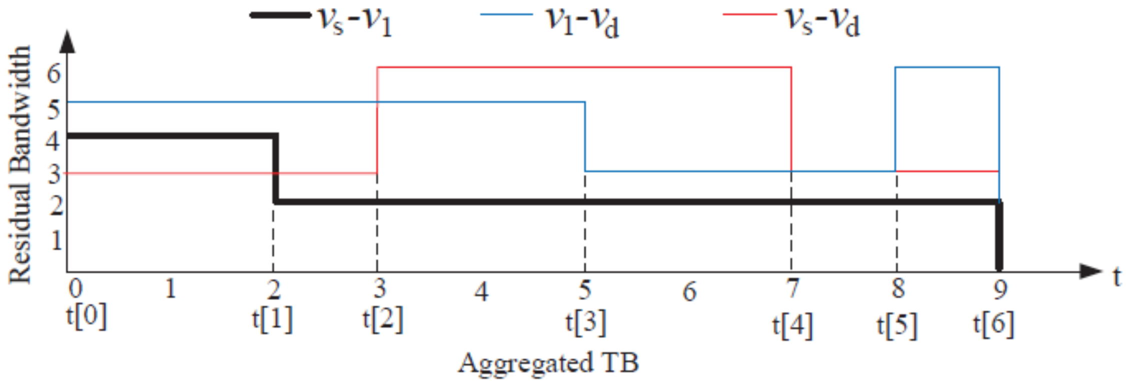


Fig. 2: An ATB list aggregating the TB of three links.

■ An aggregated time bandwidth (ATB) of all the three links is

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

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

Mathematical Model – AR

$$ar(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], p_1)$$

source node

destination node

data size to be transferred

the earliest transfer start time

the latest transfer end time (deadline)

a specified priority

- The total transfer duration of all AR requests is

$$[T^S, T^E] = [\min(t_0^S, t_1^S, ..., t_{n-1}^S), \max(t_0^E, t_1^E, ..., t_{n-1}^E)].$$

Mathematical Model – IR

$$ir(v_r^s, v_r^d, D_r, t_r^a, d_r, p_2)$$

The source node

The destination node

data size to be transferred

Random arrival time between $[T^S, T^E]$

the maximum transfer duration

a specified priority ($p_2 > p_1$)

The default transfer start time is the beginning of the next time slot to its arrival time .

Problem Definition – SAT

The **overall user satisfaction** for collaborative bandwidth scheduling of AR-IR is defined as follows:

$$SAT = \sum_{r \in \mathbb{AAR}} p_1 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]}$$

the satisfaction of each aar

$$+ \sum_{r \in \mathbb{AIR}} p_2 \cdot \frac{d_r}{[t_r^e - t_r^a] + d_r}$$

the satisfaction of each air

$$- \sum_{r \in \mathbb{PAR}} p_3 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]}. \quad (1)$$

Note: *$p_2 > p_1$, and $p_3 > p_1$. A negative satisfaction of preempted AARs which reflects a certain degree of punishment for preemption.*

Problem Definition – **BS-ARIR**

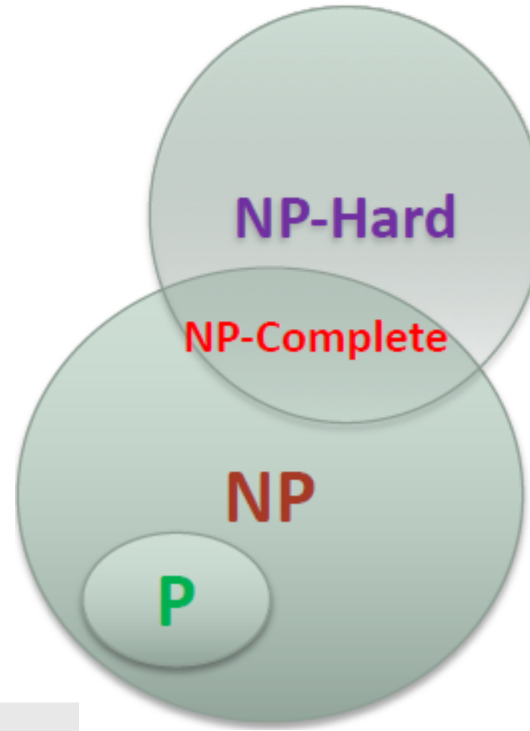
We formally define **BS-ARIR** (Bandwidth Scheduling for Advance Reservation and Immediate Reservation) as follows:

BS-ARIR Definition: Given a backbone network ***G*(*V, E*)** with an ***ATB*** list for all links and the total transfer time interval [T^{S}, T^{E}] for a batch of **AR** and **IR** requests with different priorities, our objective is to co-schedule the **AR** requests for bulk data transfer and time critical **IR** requests to maximize the overall user satisfaction as defined in ***Eq. 1***.

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

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

Proof. The decision version of BS-ARIR is as follows: Given an HPN and a set of AR and IR requests, is there a co-scheduling strategy that returns the overall user satisfaction no less than a certain ***SAT***?



- **BS-ARIR is in NP.**

Given the sets of $\mathbb{AR}$, $\mathbb{IR}$, $\mathbb{AAR}$, $\mathbb{AIR}$, and $\mathbb{PAR}$, it is easy to calculate the overall user satisfaction using Eq. 1 and compare the result with ***SAT***. We know that $\text{BS-ARIR} \in NP$.

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

■ **BS-ARIR is NP-hard.**

We prove this problem is NP-hard **by proving that a special case of this problem with a particular input structure is equivalent to a known NP-hard problem, *maxR* in [4].**

First, we consider a special case of BS-ARIR:

$$ir(v_r^s, v_r^d, D_r, t_r^a, d_r, p_2)$$

we specify a certain $t_r^a \in [T^S, T^E]$ →

$$ir(v_r^s, v_r^d, D_r, [t_r^a, t_r^a + d_r], p_2)$$

$$\Updownarrow$$

$$ar(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], p_2)$$

So, the BS-ARIR problem reduces to the problem of maximizing SAT of scheduling multiple AR requests with different priority (p_1 or p_2) in HPNs : $ar(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], p_1/p_2)$

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

Second, we consider a special case of where **all AR requests with the same priority** (i.e., $p_2 = p_1$), and no bandwidth preemption is needed. Therefore, maximizing the overall user satisfaction *SAT* reduces to maximizing *SAT'*:

$$SAT' = \sum_{r \in \mathbb{AAR}} p_1 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]}. \qquad (2)$$

Consider a special case

$$t_r^S = 0$$

Then all the AR request can be represented as:

$$ar'(v_r^s, v_r^d, D_r, [0, t_r^E])$$

Complexity Analysis - ***NP-complete*** *(4)*

Third, we further consider a particular HPN topology (Fig.3 [15]) , with a unique destination node $\boldsymbol{v}^{\mathbf{d}}$ and the bandwidth D_r/t_r^E for each link, the transfer end time of each request *ar'* on the unique path is t_r^E

So, Eq. 2 to maximize *SAT '* is further transformed to Eq. 3:

$$SAT'' = \sum_{r \in \mathbb{AAR}} p_1/2. \tag{3}$$

It is essentially equivalent to *maxR* problem in [4].

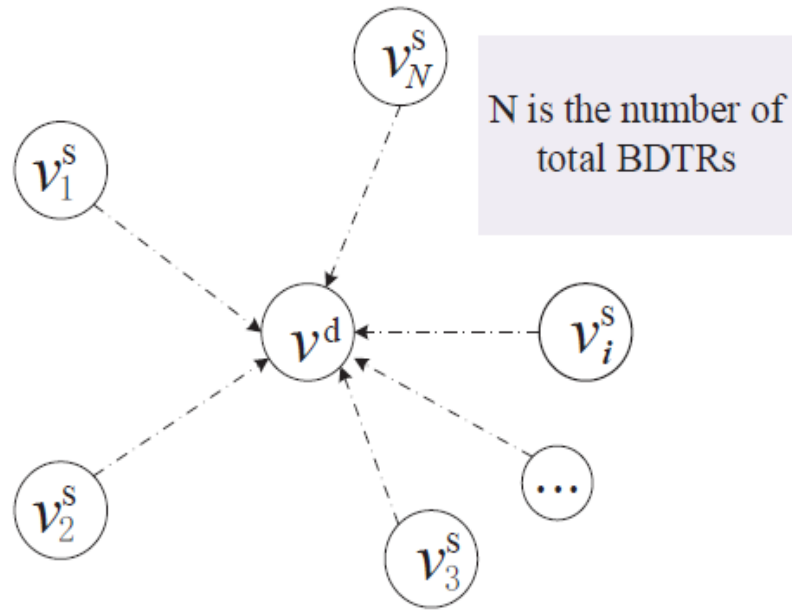


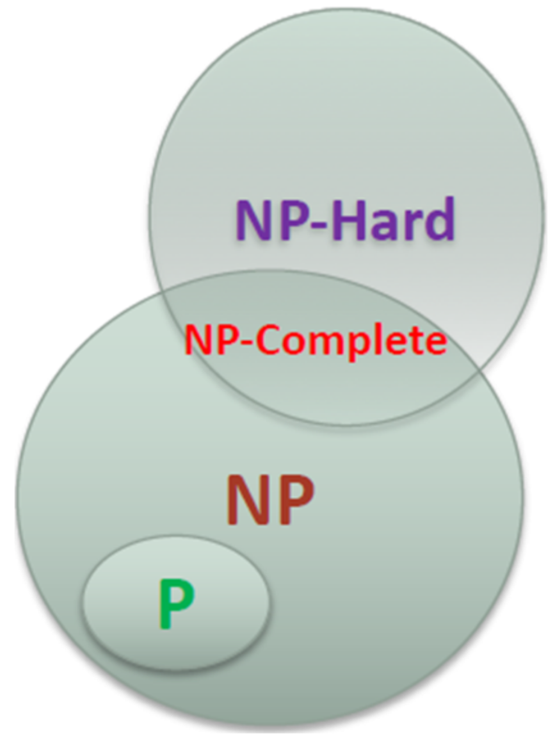
Fig. 3: An instance of a particular network structure

[4] L. Zuo, M. M. Zhu, and C. Q. Wu, "Bandwidth reservation strategies for scheduling maximization in dedicated networks," IEEE Transactions on Network and Service Management , vol. PP, no. 99, pp. 1–1, 2018.

Complexity Analysis - ***NP-complete*** *(5)*

That is to say, *maxR* problem[4] is a special case of our BS-ARIR problem. The *maxR* problem is NP-hard [4], so is our **BS-ARIR problem**.

Along with the fact that BS-ARIR is in the class of NP, we can conclude that BS-ARIR is NP-complete.



Outline

- Introduction
- Problem Formulation
- Algorithm Design
  - A. Design of Algorithm Min-R-AR
  - B. Design of Algorithm Max-S-ARIR
- Performance Evaluation
- Conclusion

Algorithm Design Overview

Phase 1: Advance Reservation for AR Requests.

For multiple AR requests, we design a periodic scheduling algorithm in advance, which is minimum resource occupy first, **Min-R-AR (Algorithm 2)**.

Phase 2: Immediate Reservation for IR Requests

During the reserved AARs transfer period, For an incoming IR request, if the bandwidth is not enough for its transfer, we design **Min-P (Algorithm 4)** with minimum preemption; And **Max-S-ARIR (Algorithm 5)** with maximum overall satisfaction for co-scheduling.

Algorithm Design for ARs (Min-R-AR)

Minimum Resource occupancy first algorithm for multiple ARs,
(i.e., first reserve a FPFB path for the AR with the minimum product of the data size and the number of path hops.)

For comparison, we also design an algorithm using the existing MBDPA algorithm in [1], referred to as **Min-BHP-AR(Algorithm 1)**

Algorithm 2 Min-R-AR(G, $\mathbb{AR}$)

Input: an HPN graph $G(V, E)$ with an ATB list of all links, and multiple ARs $(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], p_1)$ in set $\mathbb{AR}$
Output: the successfully scheduled AR set $\mathbb{AAR}$, $|AAR|$

Initialize variable $|AAR| = 0, \mathbb{AR}' = \mathbb{AR}$;
Draw the current topology G of the HPN within time interval $[T^S, T^E]$, which contains all time dots of ARs without duplicates in the increasing order;
while $\mathbb{AR}' \neq \emptyset$ **do**
  for $ar \in \mathbb{AR}'$ **do**
    $B_r^{min} = D_r / (t_r^E - t_r^S)$;
    Prune links with available bandwidth less than B_r^{min} from G to obtain graph G';
    Employ the breadth-first search algorithm to compute a path p_r with bandwidth b_r and the minimum number h_r of hops from v_r^s to v_r^d;
    if $h_r == 0$ **then**
      no path for ar;
      $\mathbb{AR}' = \mathbb{AR}' - ar$;
      continue;
    $\sigma_r = D_r \cdot h_r$;
    $ar(D_r, p_r, b_r, \sigma_r)$;
  Select the request ar with the minimum σ_r from $\mathbb{AR}'$;
  $t_r^s = t_r^S$;
  $t_r^e = t_r^s + \frac{D_r}{b_r}$;
  $\mathbb{AAR} \leftarrow ar$;
  $|AAR| = |AAR| + 1$;
  Identify the time slot intervals overlapping with $[t_r^s, t_r^e]$;
  Update the residual bandwidths of links on path p_r within time interval $[t_r^s, t_r^e]$;
  $\mathbb{AR}' = \mathbb{AR}' - ar$;
$SAT1 = \sum_{r \in \mathbb{AAR}} p_1 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]}$.
return $\mathbb{AAR}$.

Algorithm Design with Minimum preemption (Min-P(i,j,b))

Minimum preemption algorithm that identifies the AARs that can be preempted in the current time window ， to release the bandwidths for the transfer of the IR request.

Algorithm 4 Min-Preemption: $Min - P(i, j, b)$

INPUT: (i, j, b), where i and j denote the indices of two time points in $\mathbb{TP}$, and b denotes the desired amount of available bandwidths in the HPN within $[tp[i], tp[j]]$ after preemption

OUTPUT: $NULL$ or the set of AARs to be preempted

1: Identify set $\mathbb{S}_{[i,j]}$ containing existing AARs with priority value of 1 that have time interval overlapping with $[tp[i], tp[j]]$. Initialize time dot set $\mathbb{TDS} = \emptyset$, and bandwidth preemption set $\mathbb{S}'_{[i,j]} = NULL$;
2: **for** $i \le k < j$ **do**
3:    **if** $b(k) < b$ **then**
4:       Add k to $\mathbb{TDS}$;
5: **while** $|tds| > 0$ **do**
6:    Identify an existing AAR aar_m that has the same source v_r^s, same destination v_r^d and the longest time interval overlapping within $[tp[i], tp[j]]$, and maximum $\sum_{n=0}^{|tds|-1} min\,(b_m, b - b(n))$. If there are multiple existing AARs in $\mathbb{S}_{[i,j]}$ that result in the same maximum value, then choose the one with the least scheduled bandwidth;
7:    **if** no AAR is identified **then**
8:       Return $NULL$.
9:    Release the bandwidths of the links on path for aar_m, and remove aar_m from $\mathbb{S}_{[i,j]}$, add it to $\mathbb{S}'_{[i,j]}$;
10:    **for** $i \le k < j$ **do**
11:       **if** $b(k) \ge b$ **then**
12:          Remove element k from $\mathbb{TDS}$;
13: Return $\mathbb{S}'_{[i,j]}$.

Algorithm Design with Maximum Satisfaction for BS-ARIR (**Max-S-ARIR**)

First, we **call Algorithm 2** to reserve a **FPFB** path for each of the AR requests, since FPFB path can maintain continuous bandwidth for future use.

Second, during the transfer time interval [T^S, T^E] of AR requests, we calculate a **VPVB** path for an arriving IR request. If there is no sufficient bandwidth for the IR request to transfer before the deadline, the scheduler **calls Algorithm 4** to preempt some bandwidths from the AAR.

Algorithm 5 Max-S-ARIR(G,AR,IR)

Input: an HPN graph $G(V,E)$ with an ATB list of all links within $[T^S, T^E]$, and time dot priority queue tp, a set of AR$(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], pr_r)$

Output: the overall satisfaction degree SAT

Identify all time dots of ATB within time interval $[T^S, T^E]$ (including T^S and T^E), and put them in the priority queue tp in the ascending order;
$\mathbb{AAR}$ = Algorithm 2 within interval $[T^S, T^E]$;
while an IR$(v_r^s, v_r^d, D_r, t_r^a, d_r, pr_r)$ request has arrived **do**
    For the IR arriving at $t_r^a \in [T^S, T^E]$, identify the IR time interval $[tp[i], tp[i] + d_r]$ that overlaps with time interval $[tp[i], tp[j]]$, where $tp[i]$ is the time point next to t_r^a;
    Compute a series of VPVB paths within time slot $[i, j-1]$ such that each path $p_r[k]$ has the maximum bandwidth $b_r[k]$ in different time slot k, for the transfer of the arriving IR within time interval $[tp[i], tp[j]]$;
    if $\sum_{k=i}^{j-1} ((tp[k+1] - tp[k]) \cdot b_r(k)) \geq D_r$ **then**
        Add IR to $\mathbb{AIR}$, and update the residual bandwidths of links on path p_r within time interval $[i, j-1]$;
        Continue;
    Identify set $\mathbb{S}_{[i,j]}$ containing existing AARs with priority value of 1 that have time interval overlapping with $[tp[i], tp[j]]$;
    Initialize $B''_{sum} = +\infty$, $\mathbb{S}''_{[i,j']} = NULL$ and let $|s''_{[i,j']}| = +\infty$;
    for $i < j' \leq j-1$ **do**
        $D'_r = D_r - \sum_{k=i}^{j'} ((tp[k+1] - tp[k]) \cdot b_r(k))$;
        $\mathbb{S}'_{[i,j']} = MinPreemption(i, j', \frac{D'_r}{tp[j'] - tp[i]})$;
        if $\mathbb{S}'_{[i,j']} == NULL$ **then**
            Continue;
        $B'_{sum} = \sum_{k=0}^{|s'_{[i,j']}|-1} b_k$;
        if $(|\mathbb{S}'_{[i,j']}| < |\mathbb{S}''_{[i,j']}| \ || \ (|\mathbb{S}'_{[i,j']}| == |\mathbb{S}''_{[i,j']}| \ \&\& \ B'_{sum} < B''_{sum}))$; **then**
            $\mathbb{S}''_{[i,j']} = \mathbb{S}'_{[i,j']}$;
            $B''_{sum} = B'_{sum}$;
    Add IR to set $\mathbb{AIR}$, and update the residual bandwidths of links on path p_r within time interval $[i, j']$;
    Add AARs in set $\mathbb{S}''_{[i,j-1]}$ to the preempted set $\mathbb{PAR}$;
$SAT = \sum_{r \in \mathbb{AAR}} p_1 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]} + \sum_{r \in \mathbb{AIR}} p_2 \cdot \frac{d_r}{[t_r^e - t_r^a] + d_r} - \sum_{r \in \mathbb{PAR}} p_3 \cdot \frac{t_r^E - t_r^S}{[t_r^e - t_r^S] + [t_r^E - t_r^S]}$;
Return SAT.

A greedy algorithm (**Algorithm 3**) is also designed for comparison.

Outline

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

> - ***A. Simulation Setup***
> - ***B. Performance Evaluation of Min-R-AR for Multiple ARs***
> - ***C. Performance Evaluation of Max-S-ARIR Co-scheduling for AR and IR***

*Simulation Setup -- **network***

We set the total time slots to span across 20 time units, and the start time $\boldsymbol{T^S = 0}$.

The link bandwidths follow a normal distribution:

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

b^{max}: 100Gb/s

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

Simulation experiments used **six random networks** of different sizes:

TABLE II: Network sizes.

Index of network size	1	2	3	4	5	6
Number of nodes	30	60	80	100	120	150
Number of links	50	120	160	200	240	300

Simulation Setup -- ***workload***

Scheduling workloads in terms of the number of ARs/IRs

TABLE III: Scheduling workloads.

Index of workload	1	2	3	4	5	6
Number of ARs	100	200	400	600	800	1000
Number of IRs	10	20	40	60	80	100

In each run of the simulation, we randomly generate ARs and IRs:

$$ar(v_r^s, v_r^d, D_r, [t_r^S, t_r^E], p_1)$$

- v_r^s, v_r^d: two randomly selected nodes
- D_r: A random integer
- t_r^S: 0 –19
- t_r^E: 1 – 20
- p_1: 1

$$\text{IR}\ (v_r^s, v_r^d, D_r, t_r^a, d_r, p_2)$$

- t_r^a: 0 –20
- d_r: Random,<20
- p_2: 2

Performance Evaluation of ***Min-R-AR*** *for Multiple ARs*

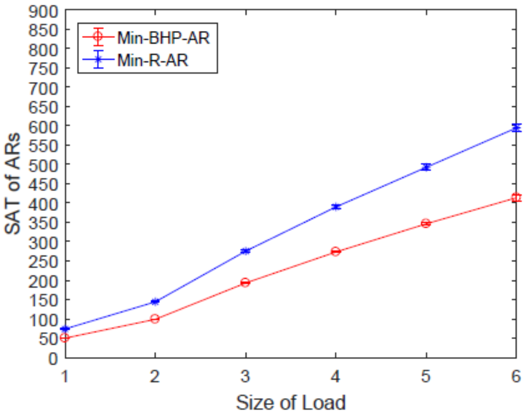


Fig. 4: In Network 1: SAT evaluation with different AR workloads.

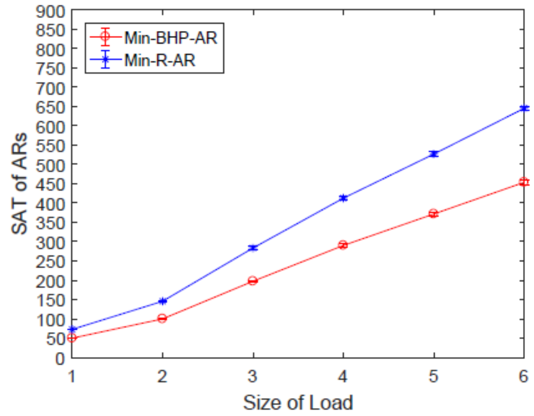


Fig. 5: In Network 2: SAT evaluation with different AR workloads.

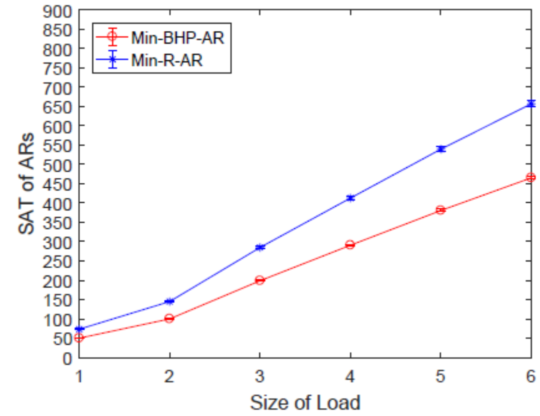


Fig. 6: In Network 3: SAT evaluation with different AR workloads.

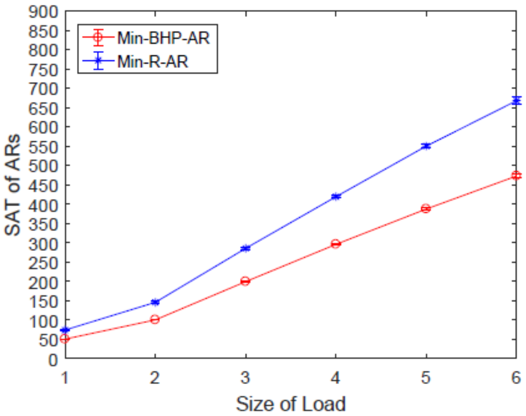


Fig. 7: In Network 4: SAT evaluation with different AR workloads.

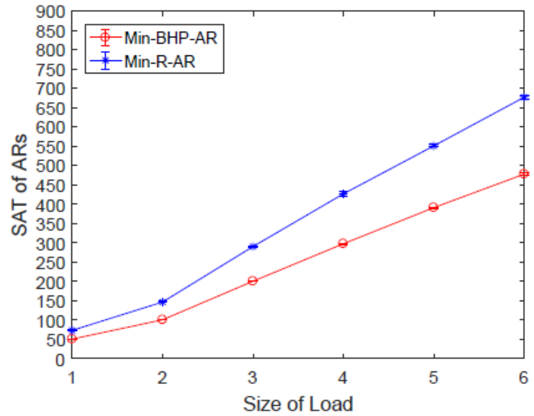


Fig. 8: In Network 5: SAT evaluation with different AR workloads.

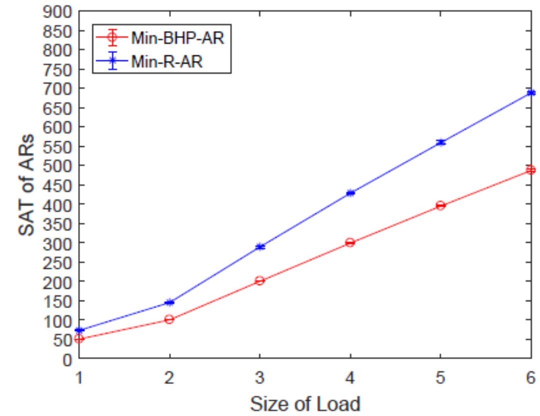


Fig. 9: In Network 6: SAT evaluation with different AR workloads.

*Performance Evaluation of **Max-S-ARIR** Co-scheduling for AR and IR*

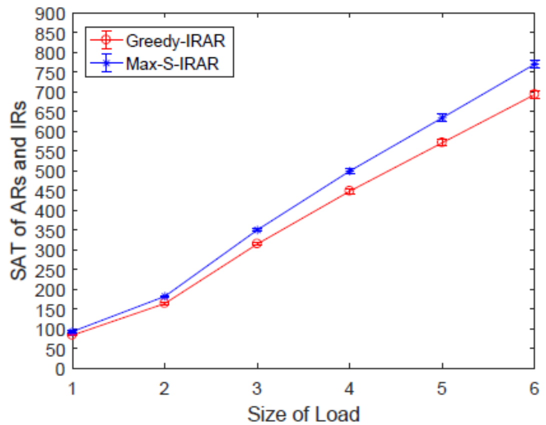


Fig. 10: In Network 1: SAT evaluation with different workloads of ARs and IRs.

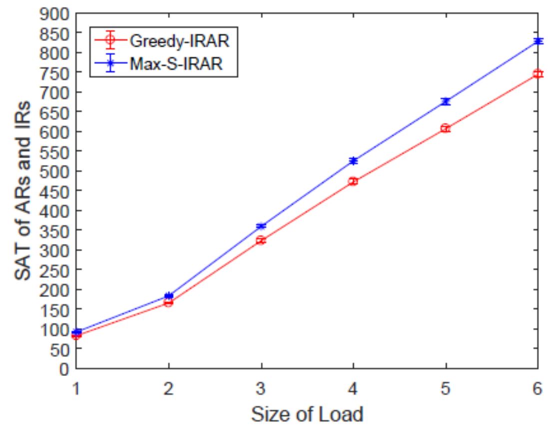


Fig. 11: In Network 2: SAT evaluation with different workloads of ARs and IRs.

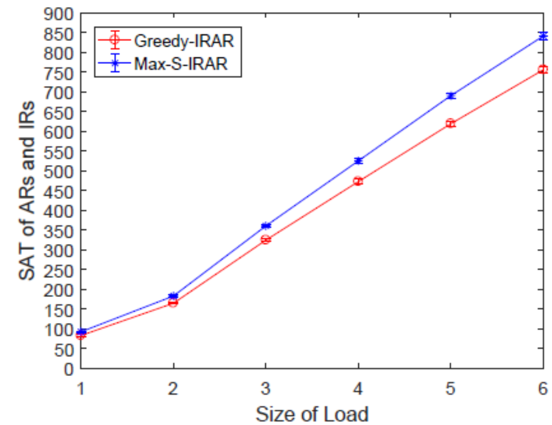


Fig. 12: In Network 3: SAT evaluation with different workloads of ARs and IRs.

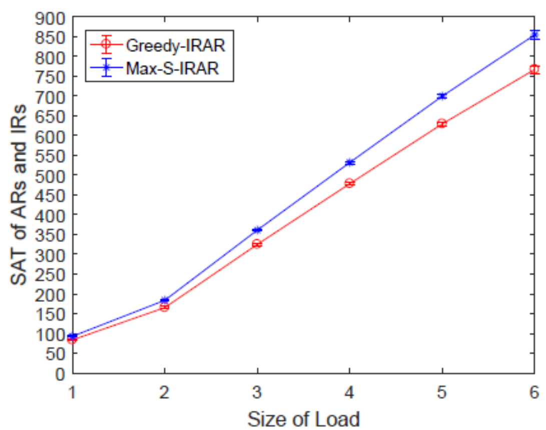


Fig. 13: In Network 4: SAT evaluation with different workloads of ARs and IRs.

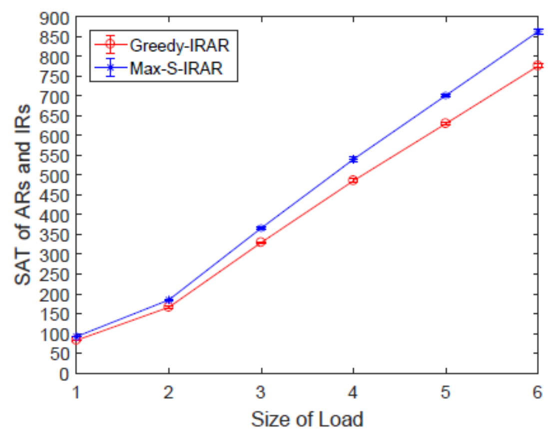


Fig. 14: In Network 5: SAT evaluation with different workloads of ARs and IRs.

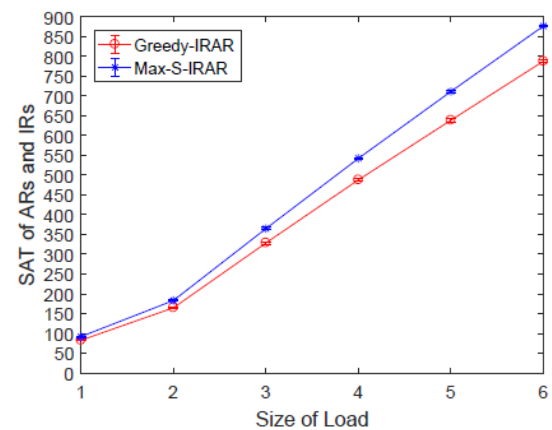


Fig. 15: In Network 6: SAT evaluation with different workloads of ARs and IRs.

Outline

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

Conclusion

- ✓ Formulated a problem of co-scheduling advance reservation and immediate reservations (**BS-ARIR)** with the objective to maximize the number of successfully scheduled requests and minimize the number of preempted advance reservation requests, while minimizing the completion time of each request.

- ✓ Proved the **NP-completeness** of **BS-ARIR**

- ✓ Proposed **heuristic algorithms** **Min-R-AR** and **Max-S-ARIR** and conducted extensive experiments, which show that the proposed algorithms significantly **outperform other existing algorithms** in terms of overall user satisfaction (SAT).

Thank you

Q & A ?

houaiqin@nwu.edu.cn