Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 84-91, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23326

Mean Based Packet Forward by Improving Bandwidth Reservation Request in HPN

J. Karthika and R. Rajesh

Department of Computer Science and Engineering,
Vivekananda Institute of Engineering and Technology for Women,
Tiruchengode, Tamilnadu

Abstract: Bandwidth reservation along paths provisioned by dedicated high-performance networks (HPNs) has proved to be a fast, reliable and predictable way to satisfy the transfer requirements of massive time-sensitive data. The problem of scheduling multiple bandwidth reservation requests (BRRs) concurrently within an HPN while achieving their best average transfer performance. Two common data transfer performance parameters are considered: the Earliest Completion Time (ECT) and the Shortest Duration (SD). Since not all BRRs in one batch can oftentimes be successfully scheduled, the problem of scheduling all BRRs in one batch while achieving their best average ECT and SD are converted into the problem of scheduling as many BRRs as possible while achieving the average ECT and SD of scheduled BRRs, respectively. Two fast and efficient heuristic algorithms with polynomial-time complexity are proposed. Extensive simulation experiments are conducted to compare their performance with two proposed naive algorithms in various performance metrics. Performance superiority of these two fast and efficient algorithms is verified. Big data Qos booster will improve the performance and it reuse the bandwidth of the sleeping node. We proved both PECT and PSD are NP-complete problems and focused on their heuristic algorithm designs. Two fast and efficient heuristic algorithms with polynomial complexity, namely FBR-ECT and FBR-SD, were proposed. FBR-ECT and FBR-SD try to concurrently schedule multiple BRRs in one batch to achieve the average ECT and SD of the scheduled BRRs.

Key words: Bandwidth Reservation · Dynamic Provisioning · High-Performance Networks · Big Data · QoS

INTRODUCTION

In extreme scale computations, big data is continuously being generated. For example, the Large Hadron Collider (LHC),1 the world's largest and most powerful particle accelerator, can generate up to 15 petabytes of data per year. Because of the challenges from the "3Vs" model of big data, i.e. increasing volume (amount of data), velocity (update time per observation) and variety (range of sources and dimension) [1], as well as the collaboration with other scientific organizations for data analysis and knowledge discovery, the data sets usually need to be transferred from the data generating center to collaborative sites located across the nation or around the globe [2-4]. Reliable data transfer is critical, especially when the data sets to be transferred are time-sensitive. However, today's default best effort network cannot meet the needs of such high-demanding data transfer since all competing data flows are treated equally and cannot provide any type of QoS. Fortunately, next-generation research and education high-performance networks (HPNs), such as the Energy Sciences Network (ESnet) [5], are developed to address such concerns. These HPNs have dedicated large bandwidth links between sites and allow data transfers to reserve bandwidth as needed in the dedicated links, thus guaranteeing predictive and reliable data transfer. For example, the big data sets generated by LHC are currently transferred from the data generating center to remote research institutions using the On-Demand Secure Circuits and Advance Reservation System (OSCARS), the bandwidth reservation service provided by ESnet.2. When a user wants to transfer data from a source end-site to a destination end-site, he/she needs to create a

---

Corresponding Author: J. Karthika, Department of Computer Science and Engineering,
Vivekananda Institute of Engineering and Technology for Women, Tiruchengode, Tamilnadu, India.

bandwidth reservation request (BRR) to be sent to the HPNs. Typically, the following information is specified in the BRR: The source end-site, the destination end-site, the data size, the maximum Local Area Network (LAN) bandwidth of the source and destination end-sites, the data available time and the data transfer deadline. The challenges of the bandwidth reservation service come from both the users and the bandwidth reservation service providers. Besides the data transfer deadline, users sometimes also want to achieve other data transfer performance parameters. For instance, two common such data transfer performance parameters are the Earliest Completion Time (ECT) and the Shortest Duration (SD)The problem of scheduling all BRRs in one batch in an HPN while achieving their best average ECT and SD. Since not all BRRs in one batch can oftentimes be successfully scheduled, the problem of scheduling all BRRs in one batch while achieving their best average ECT and SD are converted into the problem of scheduling as many BRRs as possible while achieving the average ECT and SD of scheduled BRRs, respectively. For convenience, these two converted problems are abbreviated as the Problem of Earliest Completion Time (PECT) and the Problem of Shortest Duration (PSD), respectively. We prove that both PECT and PSD are NP-complete problems. Hence, we focus on the heuristic algorithm designs. Two fast and efficient heuristic algorithms with polynomial complexity are proposed, namely Fast Bandwidth Reservation algorithm for PECT (FBR-ECT) and Fast Bandwidth Reservation algorithm for PSD (FBR-SD). For comparison purpose, we also propose two naive algorithms, namely Naive Bandwidth Reservation algorithm for PECT (NBR-ECT) and Naive Bandwidth Reservation algorithm for PSD (NBRSD). The performance superiority of FBR-ECT and FBR-SD is verified by extensive experiments on simulated ESnet topology in comparison with NBR-ECT and NBR-SD.

Mathematical Models and Notations: Suppose we have an example HPN, topology of which, G, is shown in the left side of Fig. 1. Suppose G receives a BRR at time point 0, the received BRR tries to transfer 24 Gb data from vs to vd within time interval [0, 10*s*] and the specified maximum LAN bandwidth is 8 Gb/s. It is easy to see that G consists of four nodes, namely vs, a, b and vd and four edges, namely $vs - a$, $a - vd$, $vs - b$ and $b - vd$. topology of an HPN can be modeled as a graph $G(V, E)$, where V and E represent the set of nodes and the set of edges, respectively. $V = \{vs, a, b, vd\}$ and $E = \{vs - a, a - vd, vs - b, b - vd\}$. A BRR can be represented as $(vs, vd, D, Bmax, [tS, tE])$, where vs, vd, Bmax and D denote the source node, the destination node, the maximum LAN bandwidth and the total size of data to be transferred from the earliest data transfer start time tS to the latest data transfer finish time (deadline) tE, respectively. A timewindow is defined as a time interval consisting of one timestep or more consecutive timesteps Timewindow j, denoted by twj, can be represented as $[twsj, twej]$, where $twsj$ and $twej$ denote the start time and end time of the corresponding time interval, respectively. For example, with the three timesteps in G, namely $ts0 = [0, 4s]$, $ts1 = [4s, 6s]$ and $ts2 = [6s, 10s]$ (they are also timewindows), we can derive three more time windows: [0, 6*s*], [0, 10*s*] and [4*s*, 10*s*]. Given N timesteps, the total number of time points is $(N + 1)$. The time interval between any two consecutive time points is a timestep and that between any two different time points is a timewindow. The topology of an HPN might change from time to time, QRECT and QRSD of a BRR are relative to time points. For the same BRR at different time points, its QRECT and QRSD might be different. For example, the QRECT for the BRR (24 Gb, 8 Gb/s, [0, 10*s*]) at time point 0 is ($vs - b - vd$, 6 Gb/s, [0, 4*s*]). Suppose at time point 0.5*s*, available bandwidths of both edge $vs - a$ and edge $a - vd$ within time interval [0.5*s*, 4*s*] increase from 4 Gb/s to 8 Gb/s. In this case, the QRECT for the given BRR becomes ($vs - a - vd$, 8 Gb/s, [0.5*s*, 3.5*s*]). In this paper, QRECT and QRSD of a BRR are defined as its QRECT and QRSD at the time point when the scheduling algorithm starts to process that BRR. Under themaximum LAN bandwidth constraint, the reserved bandwidth is upper limited by $Bmax$. We can derive that the minimum data transfer duration of a BRR equals D $Bmax$, denoted by $tmin$. If a BRR can be successfully scheduled within timewindow twj, the duration of twj must be no less than $tmin$, namely $(twej - twsj) \geq tmin$. For example, for BRR (24 Gb, 8 Gb/s, [0, 10*s*]), $tmin$ = 24 Gb, 8 Gb/s = 3*s*. It is easy to see that the given BRR cannot be scheduled within timewindow [4*s*, 6*s*] since its duration is (6*s* - 4*s*) = 2*s*, less than the minimum data transfer duration requirement of 3s. Removing redundant edges within a timewindow and filtering timewindows by using $tmin$ can greatly improve the overall BRR scheduling speed and efficiency [6].

Problem Formulation and Complexity Analysis: Achieving the optimal scheduling option for each BRR in one BRR batch, namely local single BRR optimality, does not lead to the best overall scheduling performance of all BRRs in the BRR batch, namely the global BRR optimality. The mathematical formulation of PECT and PSD is given after the example. We then analyze the complexity of PECT and PSD and provide the NP-complete proof [7].

Problem Formulation: For multiple BRRs in one batch, the best scenario for both the users and the bandwidth reservation service provider is that all these BRRs can be successfully scheduled and their best average data transfer performance parameters can also beachieved. Suppose we have a BRR batc *LBRR*containing multiple BRRs and a QR list *LQR* containing the QRs of the successfully scheduled BRRs.

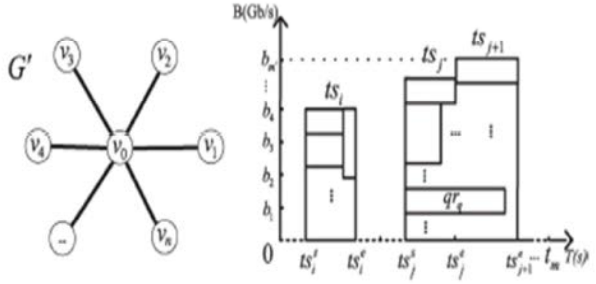


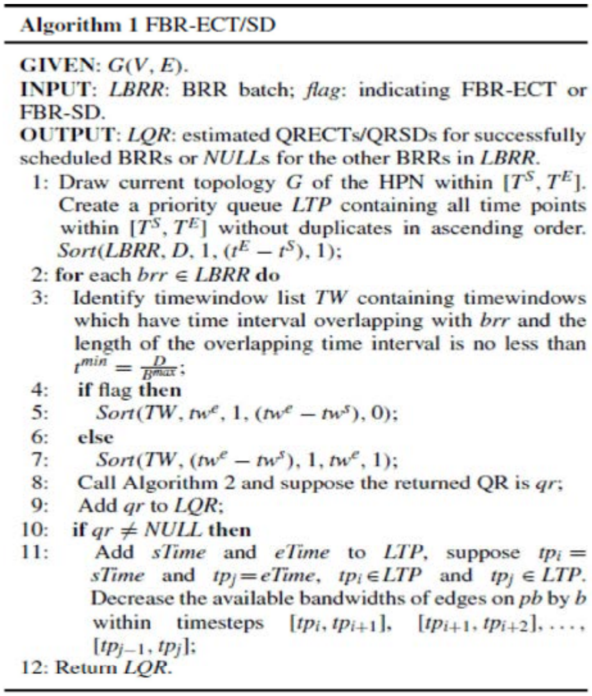
Fig. 1: Topology *G_* of an example HPN for the special simple instance of PECT/PSD (left) and the available bandwidth table of edge *v*0 - *vk* of *G*.

Problem Complexity And Analysis: We now prove that both PECT and PSD are NP-complete problems. According to [20], to prove problem *A* is NP-hard, we only need to reduce an arbitrary instance of a known NP hard problem *B* to an instance of *A* with a particular structure. Similar to the NP-hard proof in [14], we first introduce a special simple instance of PECT and PSD. Both PECT and PSD can be represented by the same special instance. For time step *tsi* on edge *v*0 - *vk*, we take out all these *bik* QRs. For each of these QRs, we assume its size equals the data size it transfers, namely the data size of the BRR for this QR. Because of the ignored QRs, *v*0 - *vk* might have different available bandwidths within time step *tsi*. We can find out all available bandwidths of *v*0 - *vk* within *tsi* in polynomial time and we assume the available bandwidth of edge *v*0 – *vk* within timestep *tsi*, *B(v*0 - *vk, tsi)*, equals the maximum value. The size of timestep *tsi* is defined as the available bandwidth of edge *v*0 - *vk* within *tsi* times the duration of *tsi*, namely *B(v*0 *- vk, tsi)* × *(tsei – tssi)*. We ignore the shape of each QR in timestep *tsi* and we only consider its size. With the special simple instance of PECT/PSD, the remaining problem is: Can we successfully place these *N* QRs within these *T* timesteps? Since we assume the BRR scheduling ratio has a higher priority over the average ECT/SD of scheduled BRRs in one batch, if we can prove the above remaining problem is NP-complete, both PECT and PSD will also be Npcomplete.

Fast and Efficient Algorithm for PECT and PCD: In this section, we focus on the heuristic algorithm design for PECT and PSD. Two fast and efficient heuristic algorithms are proposed: FBR-ECT and FBR-SD. For FBR-ECT and FBRSD, detailed algorithm designs are shown first, followed by the step-by-step explanations and illustrations by using the same example. To save space, the algorithm design and explanation of FBR-ECT and FBR-SD are merged together as FBR-ECT/SD.

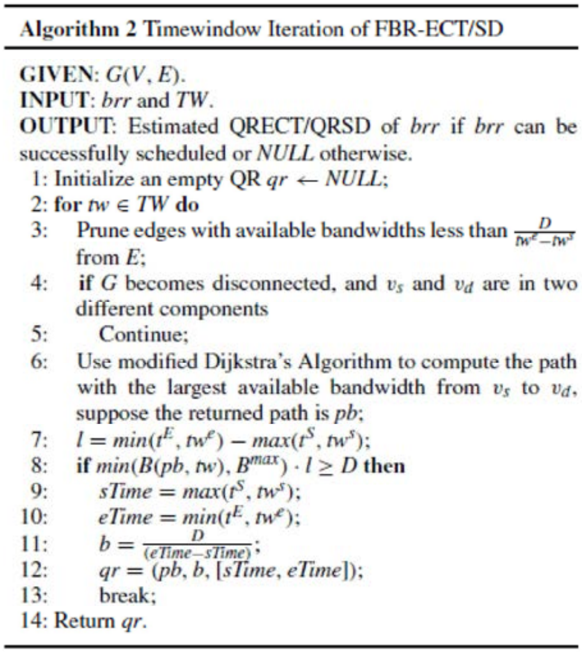
Algorithm Design: We introduce one sorting function to facilitate our algorithm design as follows [8]:

*Sort(a, b, c, d, e)***,** where *a* represents the object list to be sorted, *b* and *d* denote two sorting criteria while *c* and *e* only have two values, 0 and 1, with the meaning of sorting in descending order and ascending order, respectively. The above sorting function first sorts *a* according to *b* and *c*, then for these objects with the same sorting criterion *b*, further sorts them according to *d* and *e*. For FBR-ECT and FBRSD, detailed algorithm designs are shown first, followed by the step-by-step explanations and illustrations by using the same example. Because of the ignored QRs, *v*0 - *vk* might have different available bandwidths within timestep *tsi*. We can find out all available bandwidths of *v*0 - *vk* within *tsi* in polynomial time and we assume the available bandwidth of edge *v*0 – *vk* within timestep *tsi*, *B(v*0 - *vk, tsi)*, equals the maximum value.



Algorithm 1 FBR-ECT/SD

```
GIVEN: $G(V, E)$.
INPUT: $LBRR$: BRR batch; $flag$: indicating FBR-ECT or FBR-SD.
OUTPUT: $LQR$: estimated QRECTs/QRSDs for successfully scheduled BRRs or $NULL$s for the other BRRs in $LBRR$.
Draw current topology $G$ of the HPN within $[T^S, T^E]$. Create a priority queue $LTP$ containing all time points within $[T^S, T^E]$ without duplicates in ascending order. $Sort(LBRR, D, 1, (t^E - t^S), 1)$;
for each $brr \in LBRR$ do
    Identify timewindow list $TW$ containing timewindows which have time interval overlapping with $brr$ and the length of the overlapping time interval is no less than $t^{min} = \frac{D}{B^{max}}$;
    if flag then
        $Sort(TW, tw^e, 1, (tw^e - tw^s), 0)$;
    else
        $Sort(TW, (tw^e - tw^s), 1, tw^e, 1)$;
    Call Algorithm 2 and suppose the returned QR is $qr$;
    Add $qr$ to $LQR$;
    if $qr \neq NULL$ then
        Add $sTime$ and $eTime$ to $LTP$, suppose $tp_i = sTime$ and $tp_j = eTime$, $tp_i \in LTP$ and $tp_j \in LTP$. Decrease the available bandwidths of edges on $pb$ by $b$ within timesteps $[tp_i, tp_{i+1}]$, $[tp_{i+1}, tp_{i+2}]$, ..., $[tp_{j-1}, tp_j]$;
Return $LQR$.
```



```
Algorithm 2 Timewindow Iteration of FBR-ECT/SD
GIVEN: $G(V, E)$.
INPUT: $brr$ and $TW$.
OUTPUT: Estimated QRECT/QRSD of $brr$ if $brr$ can be successfully scheduled or $NULL$ otherwise.
 1: Initialize an empty QR $qr \leftarrow NULL$;
 2: for $tw \in TW$ do
 3:     Prune edges with available bandwidths less than $\frac{D}{tw^e - tw^s}$ from $E$;
 4:     if $G$ becomes disconnected, and $v_s$ and $v_d$ are in two different components
 5:         Continue;
 6:     Use modified Dijkstra's Algorithm to compute the path with the largest available bandwidth from $v_s$ to $v_d$, suppose the returned path is $pb$;
 7:     $l = min(t^E, tw^e) - max(t^S, tw^s)$;
 8:     if $min(B(pb, tw), B^{max}) \cdot l \geq D$ then
 9:         $sTime = max(t^S, tw^s)$;
10:         $eTime = min(t^E, tw^e)$;
11:         $b = \frac{D}{(eTime - sTime)}$;
12:         $qr = (pb, b, [sTime, eTime])$;
13:         break;
14: Return $qr$.
```

Algorithm Explanation: A step-by-step explanation of FBR-ECT/SD is provided as follows [9].

Step 1 (Line 1 of Algorithm 1): Compute the earliest data transfer start time *TS* and the latest data transfer deadline *TE* of all BRRs in *LBRR* by using Eq. (1) and (2). Draw current topology *G* of the HPN within time interval [*TS, TE*]. Create a priority queue *LTP* containing all time points within [*TS, TE*] without duplicates in ascending order. For BRR batch *LBRR*, sort all BRRs by their data size in ascending order and for those BRRs with the same data size, further sort them by their largest possible data transfer duration in ascending order.

Step 2 (Line 2 to 11 of Algorithm 1 and Algorithm 2): Iterate through the sorted BRR batch *LBRR*. For each *brr* ? *LBRR*, identify timewindow list *TW* as stated in Line 3 of Algorithm 1. If *flag* == *true*, namely current algorithm is FBR-ECT, sort timewindows in *TW* by their end times in ascending order and for timewindows with the same end time, further sort them by their durations in descending order. While if *flag* == *flase*, namely current algorithm is FBR-SD, sort timewindows in *TW* by their durations in ascending order and for timewindows with the same duration, further sort them by their end times in ascending order. With the sorted timewindow list *TW*, call Algorithm 2 to compute the estimated QRECT/QRSD of *brr*. Suppose the returned estimated QRECT/QRSD of *brr* is *qr*, add *qr* to *LQR*. If *qr* _= *NULL*, update the available bandwidths of edges on the data transfer path by performing the operation stated in Line 11 of Algorithm 1.

Step 3 (Line 12 of Algorithm 1): Return the QR list *LQR*. As for FBR-ECT/SD, multiple strategies are used specifically to improve the overall BRR scheduling performance.

1.BRR sorting stated in Line 1 of Algorithm 1. This strategy improves BRR scheduling ratio since BRR with the least bandwidth resource is to be scheduled at first. For BRRs requiring the same bandwidth resource, the one with the shortest largest possible data transfer duration is to be scheduled first since the BRRs with longer largest possible data transfer durations have higher probabilities to be scheduled.

2.QR computation stated in Line 9 to 12 of Algorithm 2. Whenever a path can provide a valid QR within a certain interval, we would try to expand the data transfer interval as much as possible.

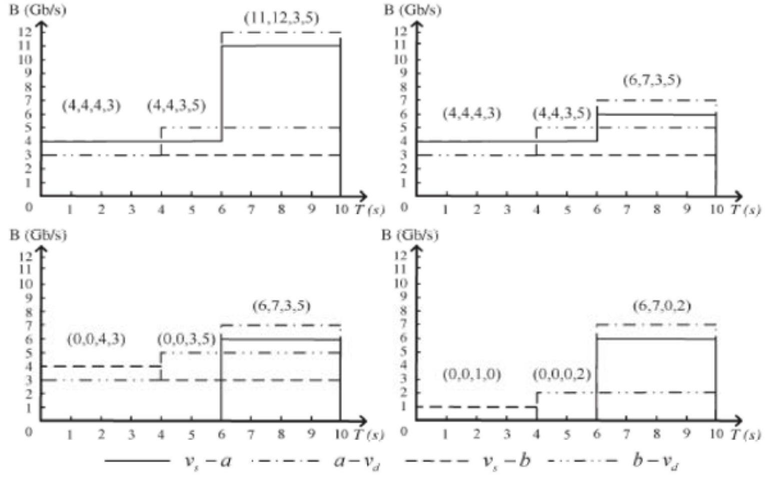


Fig. 2: Available bandwidth table of edges in G after scheduling brr0, brr2, brr1 and brr3 by using FBR-ECT.

3.Timewindow filtering stated in Line 3 of Algorithm 1 and edge pruning stated in Line 3 to 5 of Algorithm 2. These two strategies shrink the cardinality of *TW* and prune redundant edges from *G* within a timewindow, which reduce the number of timewindows to be iterated and improve the path searching speed by using modified Dijkstra's Algorithm.

4.Timewindows sorting stated in Line 4 to 7 of Algorithm 1 and timewindow list *TW* iteration breaking stated in Line 13 of Algorithm 2. For FBR-ECT, the timewindow sorting expands the data transfer interval of a BRR as much as possible while keeping its data transfer end time as early as possible. For FBR-SD, the timewindow sorting tries to finish the data transfer as early as possible while keeping the data transfer duration as short as possible. If one valid QR is found within a timewindow, iteration of TW breaks, which improves the overall BRR scheduling Speed.

Algorithm Illustration Of FBR-ECT: We use the following list of BRRs to illustrate FBR-ECT (the input parameter *flag* == *true*) on the topology of the example HPN shown in Fig. 1.

*brr*0 = {12 Gb,6 Gb/s, [0, 5*s*]}

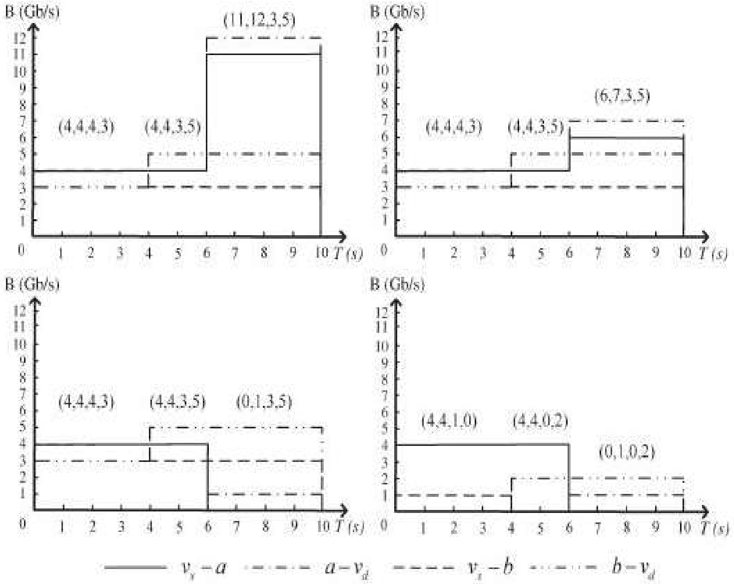
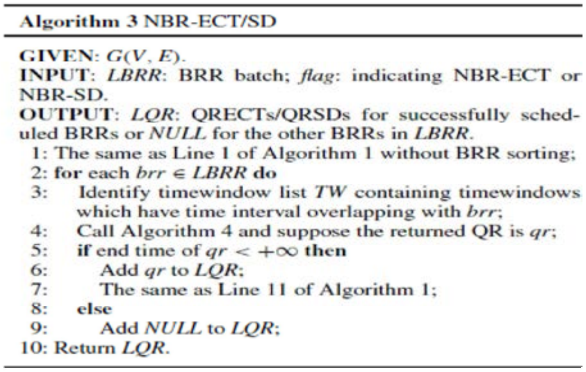


Fig. 3: Available bandwidth table of edges in G after scheduling brr0, brr2, brr1 and brr3 by using FBR-SD.

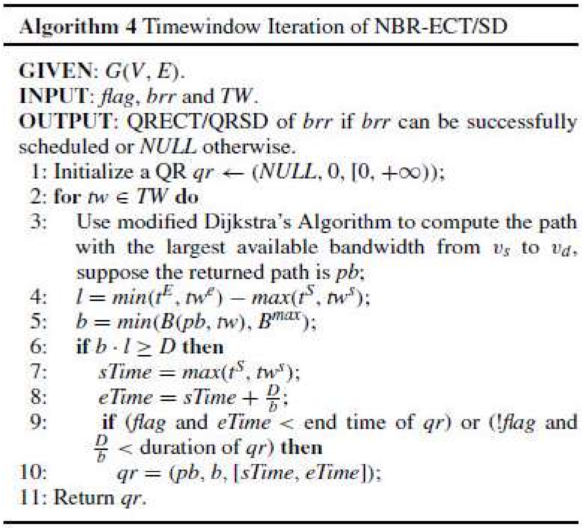
Naive Algorithm Design for PECT and PSD: Since there are currently no existing algorithms to achieve the same objectives as PECT and PSD, for comparison purpose, two naive heuristic algorithms, namely NBR-ECT and NBR-SD, are proposed in this section. Similar to FBR-ECT and FBR-SD, the algorithm design and explanation of NBR-ECT and NBR-SD are merged together as NBR-ECT/SD.

Algorithm Design: NBR-ECT/SD uses traditional bandwidth scheduling strategy [10] to schedule BRRs in one BRR batch: Schedule these BRRs one by one and try to achieve QRECT/QRSD for each BRR to achieve the average ECT/SD of the scheduled BRRs. For a BRR, NBR-ECT/SD identifies and returns its actual QRECT/QRSD, which is different from FBR-ECT/SD,two naive heuristic design, namely NBR-ECT and NBR-SD, are proposed in this section. Similar to FBR-ECT and FBR-SD, the algorithm design and explanation of NBR-ECT and NBR-SD are merged together as NBR-ECT/SD [11].



Algorithm 3 NBR-ECT/SD

```
GIVEN: G(V, E).
INPUT: LBRR: BRR batch; flag: indicating NBR-ECT or NBR-SD.
OUTPUT: LQR: QRECTs/QRSDs for successfully scheduled BRRs or NULL for the other BRRs in LBRR.
The same as Line 1 of Algorithm 1 without BRR sorting;
for each $brr \in LBRR$ do
    Identify timewindow list TW containing timewindows which have time interval overlapping with brr;
    Call Algorithm 4 and suppose the returned QR is qr;
    if end time of $qr < +\infty$ then
        Add qr to LQR;
        The same as Line 11 of Algorithm 1;
    else
        Add NULL to LQR;
Return LQR.
```



Algorithm 4 Timewindow Iteration of NBR-ECT/SD

```
GIVEN: $G(V, E)$.
INPUT: $flag$, $brr$ and $TW$.
OUTPUT: QRECT/QRSD of $brr$ if $brr$ can be successfully scheduled or $NULL$ otherwise.
1: Initialize a QR $qr \leftarrow (NULL, 0, [0, +\infty))$;
2: for $tw \in TW$ do
3:     Use modified Dijkstra's Algorithm to compute the path with the largest available bandwidth from $v_s$ to $v_d$, suppose the returned path is $pb$;
4:     $l = min(t^E, tw^e) - max(t^S, tw^s)$;
5:     $b = min(B(pb, tw), B^{max})$;
6:     if $b \cdot l \geq D$ then
7:         $sTime = max(t^S, tw^s)$;
8:         $eTime = sTime + \frac{D}{b}$;
9:         if ($flag$ and $eTime <$ end time of $qr$) or ($!flag$ and $\frac{D}{b} <$ duration of $qr$) then
10:            $qr = (pb, b, [sTime, eTime])$;
11: Return $qr$.
```

Algorithm Explanation: A step-by-step explanation of NBR-ECT/SD is provided as follows.

Step 1 (Line 1 of Algorithm 3): This step is the same as Step 1 of the Algorithm Explanation of FBR-ECT/SD without the BRR sorting procedure.

Step 2 (Line 2 to 9 of Algorithm 3 and Algorithm 4): Iterate through BRR batch *LBRR*. For each *brr* ? *LBRR*, identify timewindow list *TW* containing timewindows which have time interval overlapping with *brr*. Call Algorithm 4 to compute the QRECT/QRSD of *brr*. Suppose the returned QRECT/QRSD of *brr* is *qr*. If end time of *qr* < +8, add *qr* to *LQR* and then update the available bandwidths of edges on the data transfer path by performing the operation stated in Line 11 of Algorithm 1; otherwise, add *NULL* to *LQR*.

Algorithm 4: Iterate through the timewindow list *TW*. For each *tw* **?** *TW*, use modified Dijkstra's Algorithm to find the path with the largest available bandwidth from *vs* to *vd*. We then check if the returned path can finish the data transfer of *brr* within the time interval overlapped between *tw* and *brr* and provide a valid QR. If *flag* == *true*, QRECT among QRs provided by all timewindows in *TW* is recorded; otherwise, QRSD is recorded. After iteration of *TW*, return the recorded QR *qr*.

Step 3 (Line 10 of Algorithm 3): Return the QR list LQR.

Module Description

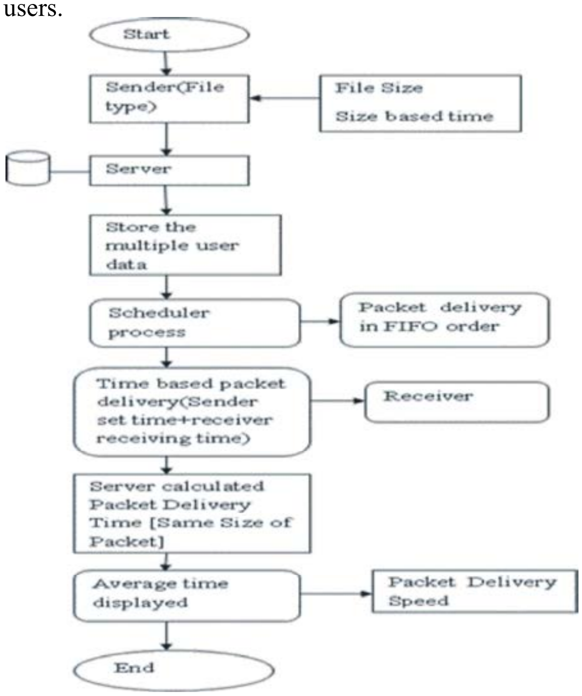
Bandwidth Creation Model: Large-scale experimental and computational scientific applications, big data is being generated on a daily basis. Such large volumes of data usually need to be transferred from the data generating center to remotely located scientific sites for collaborative data analysis in a timely manner. Bandwidth reservation along paths provisioned by dedicated High-Performance Networks (HPNs) has proved to be a fast, reliable and predictable way to satisfy the transfer requirements of massive time-sensitive data. model of big data, i.e. increasing volume (amount of data), velocity (update time per observation) and variety (range of sources and dimension), as well as the collaboration with other scientific organizations for data analysis and knowledge discovery, the data sets usually need to be transferred from the data generating center to collaborative sites located across the nation or around the globe. Reliable data transfer is critical, especially when the data sets to be transferred are time-sensitive. However, today's default best effort network cannot meet the needs of such high-demanding data transfer since all competing data flows are treated equally and cannot provide any type of Quality of Service (QoS) [12].

Data Transferring Time Builder: Bandwidth reservation along paths provisioned by dedicated High-Performance Networks (HPNs) has proved to be a fast, reliable and predictable way to satisfy the transfer requirements of massive time-sensitive data model of big data, i.e. increasing volume (amount of data), velocity (update time per observation) and variety (range of sources and dimension), as well as the collaboration with other scientific organizations for data analysis and knowledge discovery, the data sets usually need to be transferred from the data generating center to collaborative sites located across the nation or around the globule. In such HPNs, when a user wants to transfer data from a source end-site to a destination end-site, he/she needs to create a Bandwidth Reservation Request (BRR) [13] to be sent to the HPNs. Typically, the following information is specified in the BRR: The source end-site, the destination end-site, the data size, the maximum Local Area Network (LAN) bandwidth of the source and destination end-sites, the data available time and the data transfer deadline.

Transmission and Time Controller: The challenges of the bandwidth reservation service come from both the users and the bandwidth reservation service providers. Besides the data transfer deadline, users sometimes also want to achieve other data transfer performance parameters. For instance, two common such data transfer performance parameters are the Earliest Completion Time

(ECT) and the Shortest Duration (SD), While from the bandwidth reservation service providers' perspective, all BRRs in one batch should be scheduled concurrently in the HPNs for high bandwidth resource utilization and throughput purposes. To reduce the variance of the computed user-predefined data transfer performance parameters, such as ECT and SD, the best average data transfer performance of all BRRs in one batch should be achieved, namely the global BRR optimality [14].

Speed Monitoring Model: Bandwidth scheduling method [15] can be categorized into two groups, Instant scheduling, which schedules one single BRR at a time and Periodical scheduling which schedules multiple BRRs accumulated with in a time period at a time. For example, the scheduling algorithms proposed in are instant scheduling algorithms. A periodical scheduling problem is normally more difficult than an instant scheduling problem since its subject is multiple BRRs instead of one single BRR.. However, a periodical scheduling algorithm may schedule these multiple BRRs in some order other than the BRRs receiving order, which could lead to a better scheduling performance than the "First In, First Out" (FIFO) strategy used by an instant scheduling algorithm. Zuo *et al*. studied the problem of scheduling multiple BRRs with different priorities in an HPN. In the study, two optimal algorithms are proposed and for each BRR, the proposed algorithms identify and return the bandwidth reservation options with ECT and SD to the users.



Performance Evaluation: In this section, we run extensive simulations to compare the overall performance of the two proposed algorithms for PECT, namely FBR-ECT and NBR-ECT and those for PSD, namely FBR-SD and NBR-SD. To mimic the real ESnet scenario and fully compare these proposed algorithms, we conduct our experiments on simulated ESnet, whose topology is drawn by using the real data gathered from Esnet3.

Conclusion and Future Work: In this paper, we studied the problem of scheduling all BRRs in one BRR batch while achieving their best average data transfer performance within HPNs. Two common data transfer performance parameters, ECT and SD, were considered. Since not all BRRs in one batch can oftentimes be successfully scheduled, the problem of scheduling all BRRs in one batch while achieving their best average ECT and SD in one BRR batch were converted into the problem of scheduling as many BRRs as possible while achieving the average ECT and SD of scheduled BRRs, which were abbreviated as PECT and PSD, respectively. For performance comparison purpose, we also proposed two naive heuristic algorithms, namely NBRECT and NBR-SD and these two algorithms try to achieve the QRECT and QRSD of each BRR in one BRR batch to achieve the average ECT and SD of the scheduled BRRs. For these four algorithms, detailed algorithm designs, explanations and illustrations using the same example were provided. In the future, we plan to collaborate with network service providers to implement and integrate our efficient and novel scheduling algorithm in a real network such as ESnet and GENI for performance evaluation purposes. We also plan to extend our work to data transfer and bandwidth scheduling in Cloud Computing environment.

REFERENCES

1. Feldman, D., M. Schmidt and C. Sohler, 2013. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering, in Proc. 24th Annu. ACM-SIAM Symp. Discr. Algorithms, pp: 1434–1453.
2. Shu, T., C. Wu and D. Yun, 2013. Advance bandwidth reservation for energy efficiency in high-performance networks, in Proc. IEEE 38th Conf. Local Comput. Netw., pp: 541 548.
3. Zuo, L., M. Khaleel, M. Zhu and C. Wu, 2013. On fixed-path variablebandwidth scheduling in high-performance networks, in Proc. IEEE Int. Conf. Green Comput. Commun., pp: 23-30.

4. Rao, N., *et al*., 2006. Control plane for advance bandwidth scheduling in ultra high-speed networks, in Proc. 25th IEEE Int. Conf. Comput. Commun., pp: 1-5.
5. Charbonneau, N., V.M. Vokkarane, C. Guok and I. Monga, 2011. Advance reservation frameworks in hybrid IP-WDM networks, IEEE Commun. Mag., 49(5): 132-139.
6. Zheng, X., M. Veeraraghavan, N. Rao, Q. Wu and M. Zhu, 2005. CHEETAH: Circuit-switched high-speed end-to-end transport architecture testbed, IEEE Commun. Mag., 43(8): 11-17.
7. Summerhill, R., 2006. The new Internet2 network, presented at the 6th Global Lambda Integrated Facility, Prague, Czech Republic.
8. Lehman, T., J. Sobieski and B. Jabbari, 2006. DRAGON: A framework for service provisioning in heterogeneous grid networks, IEEE Commun. Mag., 44(3): 84-90.
9. Balman, M., E. Chaniotakisy, A. Shoshani and A. Sim, 2010. A flexible reservation algorithm for advance network provisioning, in Proc. ACM/IEEE Int. Conf. High Perform. Comput., Netw., Storage Anal., pp: 1-11.
10. Lin, Y. and Q. Wu, 2013. Complexity analysis and algorithm design for advance bandwidth scheduling in dedicated networks, IEEE/ACM Trans. Netw., 21(1): 14-27.
11. Zuo, L., M. Zhu and C. Wu, 2013. Fast and efficient bandwidth reservation algorithms for dynamic network provisioning, J. Netw. Syst. Manage., pp: 1-25.
12. Sharma, S., D. Katramatos and D. Yu, 2011. End-to-end network QoS via scheduling of flexible resource reservation requests, in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., pp: 1-10.
13. Sharma, S., D. Katramatos, D. Yu and L. Shi, 2012. Design and implementation of an intelligent end-to-end network QoS system, in Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal., pp: 1-11.
14. Recio, J., E. Grasa, S. Figuerola and G. Junyent, 2005. Evolution of the user controlled lightpath provisioning system, in Proc. 7th Int. Conf. Transp. Opt. Netw., 1: 263-266.
15. Sahni, S., *et al*., 2007. Bandwidth scheduling and path computation algorithms for connection-oriented networks, in Proc. 6th Int. Conf. Netw., pp: 47-47.