SCIENCE CHINA
Information Sciences

• **RESEARCH PAPER** •

February 2016, Vol. 59 022310:1–022310:14
doi: 10.1007/s11432-015-5378-7

Virtual network embedding for hybrid cloud rendering in optical and data center networks

Weigang HOU[1,2,3] & Lei GUO[1]*

[1] *School of Computer Science and Engineering, Northeastern University, Shenyang* 110819*, China;*
[2] *State Key Laboratory of Networking and Switching Technology, Beijing* 100876*, China;*
[3] *State Key Laboratory of Information Photonics and Optical Communications, Beijing* 100876*, China*

Received February 13, 2015; accepted March 30, 2015; published online January 5, 2016

Abstract Animation rendering consumes massive computation time, therefore cloud rendering is emerging as a solution. Cloud rendering runs over the Data Center Network (DCN) and consolidates heterogeneous DC resources into a single cloud *renderfarm*, where plentiful computing resources can sufficiently accelerate any rendering process. And if one user wants to get a quick animation result, a high-speed optical interconnection is an urgent requirement, thus cloud rendering needs a convergence of Optical and DCN (ODCN) as the substrate network. In the ODCN supporting cloud rendering, each rendering task will be successfully handled only when we embed its virtual network into the cloud renderfarm. But because a virtual network includes virtual machines and virtual lightpaths, we must simultaneously perform the node-level mapping between virtual machine and server, as well as link-level mapping between virtual lightpath and fiber link(s). In addition, the joint implementation of the Photorealistic cloud Rendering (PR) and Non-Photorealistic cloud Rendering (NPR) should be considered to exhibit the unique animation effect with the low mapping cost. In this paper, considering the unique characteristic of hybrid cloud rendering, we flexibly select routing strategies according to the rendering task type. We then utilize server consolidation and traffic grooming to achieve node- and link-level mappings, respectively, thus building a mapping-cost-aware cloud renderfarm that includes multiple virtual networks. The mathematical formulation is also made with a bound analysis. Especially for the lower bound, we analyze the least number of servers and wavelengths (i.e., mapping cost) consumed by hybrid cloud rendering. In terms of heuristics, according to the processing order of rendering tasks, Smaller Virtual Resource First (SVRF) and Manycast Routing First (MRF) algorithms are proposed by us. In SVRF, NPR tasks are first tackled and then PR tasks follow. MRF is a reverse process of SVRF. The simulation results demonstrate the effectiveness of our methods in reducing the mapping cost because the heuristic solution well matches the lower bound.

Keywords hybrid cloud rendering, optical and data center network, virtual network embedding, mapping cost, lower bound

Citation Hou W G, Guo L. Virtual network embedding for hybrid cloud rendering in optical and data center networks. Sci China Inf Sci, 2016, 59(2): 022310, doi: 10.1007/s11432-015-5378-7

1 Introduction

Animation rendering transforms a digital design into a vivid cartoon [1]. Since every cartoon scene usually includes thousands of geometric models, it consumes such massive computation time that a stand-alone

* Corresponding author (email: guolei@ise.neu.edu.cn)

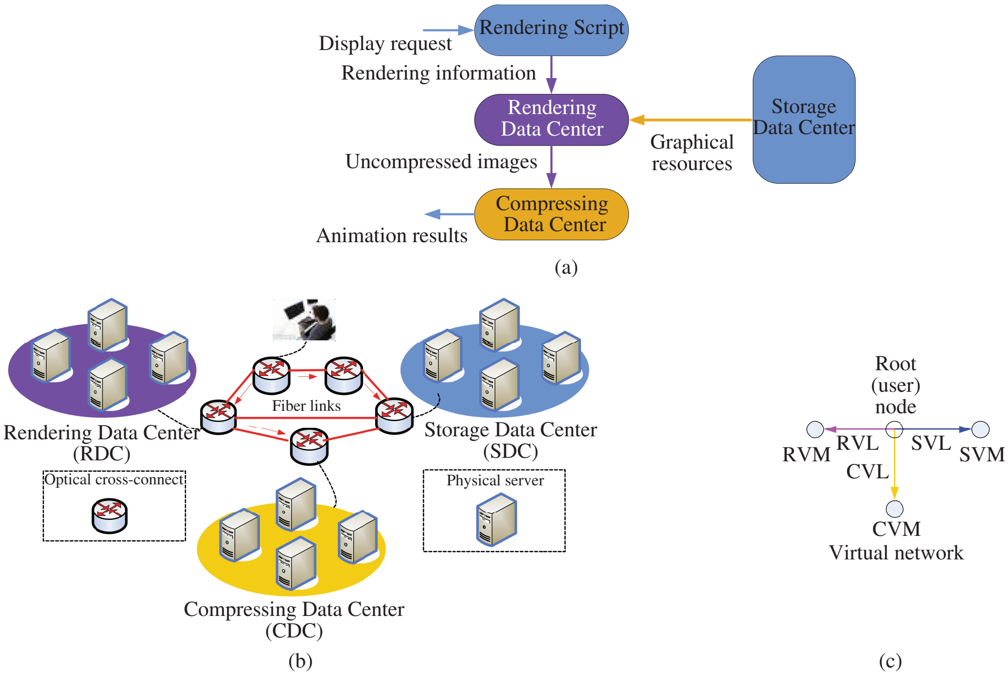


Figure 1 (Color online) (a) Cloud rendering procedure; (b) substrate ODCN supporting cloud rendering; (c) virtual network of a rendering task

rendering will spend two hours per frame. Especially for a 160-min animation, the design cycle might increase to 2.88×10^6 h.

The cluster rendering [1] lets large batches of servers simultaneously render a single file in order to shorten the design cycle. But this parallel operation might lead to high operating cost and massive idle computing resources. Consequently, cloud rendering was proposed [2,3]. In cloud rendering, heterogeneous Data Center (DC) resources are consolidated into a single cloud renderfarm. Based on the cloud computing concept, an owner manages a powerful computing program at the remote cloud renderfarm, and receives rendering tasks submitted by any user. Cloud renderfarm quickly executes the submitted task and then returns the animation result back to the user terminal. Without purchasing any expensive rendering devices with limited computational ability, a user merely pays short-term rental fees to the owner of the cloud renderfarm, where plentiful computing resources can be shared for accelerating any rendering process.

However, it is still far from the practical demands of a perfect cloud rendering. An entire cloud rendering procedure is shown in Figure 1(a). Obviously, the cloud renderfarm will run over rendering, compressing and storage DCs, which can be seen in Figure 1(b). In addition, due to space limitation, especially for cooperation across different regions, these DCs are far away from each other. Thus if one user wants to get a quick animation result, a high-speed optical interconnection is an urgent requirement for data transfer between user and designated DC(s). More importantly, some cloud rendering applications, e.g., 3D game, are very delay-sensitive, and the order execution delay mainly depends on the communication bandwidth. In order words, these applications are also bandwidth-hungry. For example, in Japan, broadband access has reached up to 100 Mbps to satisfy these delay-sensitive and bandwidth-hungry cloud rendering applications. Therefore, a perfect cloud rendering urgently needs a convergence of Optical and Data Center Network (ODCN) as the substrate network.

In the ODCN supporting cloud rendering, each rendering task makes a request of the owner to embed its virtual network into the cloud renderfarm. As shown in Figure 1(c), each rendering task can be represented by a tree-based virtual network, where the leaf node denotes the requirement of virtual resource (i.e., Virtual Machine, VM), the root node denotes the user node, and the link represents the requirement of virtual optical bandwidth (i.e., Virtual Lightpath, VL). So, every rendering task will be successfully handled only when we simultaneously perform the mapping between VM and server (i.e., node-level mapping) as well as another mapping between VL and fiber link(s) (i.e., link-level mapping). After executing all mapping processes, a cloud renderfarm will include a set of virtual networks whose virtual resources have been successfully mapped into the substrate ODCN.

Currently for virtual network embedding, most solutions are not ODCN-oriented and neglect the unique characteristics of cloud rendering [4–10]. As forerunners, we focus on the ODCN-level virtual network embedding problem with the consideration of cloud rendering features. In this paper, we propose a framework to solve this problem. Note that, the inter-DC communication is not within the scope of this paper. Our contributions are summarized as follows.

- We designed a mathematical model to formulate the ODCN-level virtual network embedding problem with the consideration of cloud rendering features.
- The NP-completeness of our problem was demonstrated and two efficient heuristics were designed to solve it within a short computation time.
- We analyzed the lower bound of the mapping cost (i.e., the least number of servers and wavelengths to be consumed), which demonstrates the effectiveness of heuristics since the mapping cost of each heuristic falls into the range of bound in simulations.

The rest of this paper is organized as follows. In Section 2, we give an overview of our framework. Based on the proposed framework, we formulate our problem with a lower bound in Section 3, and propose heuristics in Section 4. We present simulation results in Section 5. Finally, in Section 6, we introduce some existing solutions of virtual network embedding, before concluding the paper in Section 7.

2 Overview of our framework

In our framework, we first introduce the unique characteristics of a hybrid cloud rendering approach. Considering these characteristics, we perform the flexible routing decision before running ODCN-level virtual network embedding.

2.1 Hybrid cloud rendering

In addition to Photorealistic cloud Rendering (PR) that can create a real-world animation effect, we also need Non-Photorealistic cloud Rendering (NPR) to exhibit the unique animation effect by using other low-cost art forms, such as Chinese ink painting and paper cutting. Obviously, this hybrid cloud rendering approach reduces design cycle and resource consumption, because NPR approach creates a refreshing feeling for audiences at a relatively low cost. The hybrid cloud rendering, however, brings new characteristics: (1) there exists a resource-requirement gap between PR and NPR tasks, i.e., a PR task consumes more resources compared with an NPR task; (2) an NPR task can be handled by any DC, while a PR task must be tackled by rendering, compressing, and storage DCs. Thus for each NPR task, the corresponding virtual network is only a single branch including one root node (i.e., user) and one leaf node (i.e., the VM to be mapped into a random DC). While for each PR task, the corresponding virtual network has a tree-based structure as shown in Figure 1(c).

2.2 Flexible routing decision

The aforementioned characteristics of hybrid cloud rendering motivated us to flexibly select routing strategies according to the rendering task type. More specifically, we utilize manycast routing for every PR task, because all three kinds of DCs tackle this type of task. While for every NPR task, we utilize anycast routing, because an NPR user will not be concerned about the exact location of the DC to complete this type of task, as long as both node- and link-level mappings are achieved.

2.3 Virtual network embedding

During the process of node-level mapping, we try to pack as many VMs as possible into a single server provided that this server has enough available space, i.e., server consolidation [11]. By using server consolidation, we can reduce the cost of node-level mapping (i.e., the number of consumed servers).

Note that, as shown in Figure 1(c), every PR task has three types of VMs, i.e., Rendering VM (RVM), Compressing VM (CVM), and Storage VM (SVM). Each kind of VM should be consolidated into the server with the same attribute, for example, RVM should be consolidated into a server within Rendering Data Center (RDC).

During the process of link-level mapping, the traffic grooming [12–14] is a feasible solution for the reduction of link-level mapping cost (i.e., the number of consumed wavelengths). By using traffic grooming, multiple VLs can be mapped into the same wavelength of a single fiber link. As a result, this wavelength will be shared by these VLs, because one wavelength capacity is much bigger than the virtual optical bandwidth of a VL.

3 Problem statement

In this section, we first introduce the system model, and then describe our problem formulation.

3.1 System model

The substrate ODCN-level animation rendering infrastructure includes a set of Optical Cross-Connect (OXC) nodes N, a set of fiber links E, and a set of DCs D. For the problem tractability and the analysis of the problem lower bound that will be presented later, we consider that $|N| > |D| = 3$, i.e., only three kinds of DCs (rendering, compressing and storage DCs) locate at the edge of optical backbone, and each of them connects an OXC. But this simplified model can be well extended to a case where the cardinality of the DC set D is very large as long as three kinds of DCs are all involved. We also consider that the servers have the same capacity SC, and wavelengths have the same capacity WC. Every OXC node has enough transceivers. According to the characteristics of hybrid cloud rendering, we generate two types of rendering tasks, PR and NPR.

As discussed above, the virtual network of each NPR task is represented by a 3-tuple branch $\langle s, t_1, r_1\rangle$, where s is the user node, t_1 is the virtual optical bandwidth of an NPR VL, and r_1 is the size of an NPR VM. Because an NPR user will not be concerned about the exact location of the DC to complete this task, the DC information is unknown in advance.

The virtual network of each PR task is represented by a 4-tuple tree $\langle s, D, t_2, r_2\rangle$, where the multi-dimension vector t_2 includes the virtual optical bandwidth rvl_{t_2} of the PR Rendering VL (RVL) connecting the PR RVM to be mapped into RDC, the virtual optical bandwidth cvl_{t_2} of the PR Compressing VL (CVL) connecting the PR CVM to be mapped into Compressing Data Center (CDC), and the virtual optical bandwidth svl_{t_2} of the PR Storage VL (SVL) connecting the PR SVM to be mapped into Storage Data center (SDC), which can be seen in Figure 1(c); the multi-dimension vector r_2 includes the size rvm_{r_2} of a PR RVM, the size cvm_{r_2} of a PR CVM, and the size svm_{r_2} of a PR SVM; We consider that t_1, r_1, rvl_{t_2}, cvl_{t_2}, svl_{t_2}, rvm_{r_2}, cvm_{r_2}, and svm_{r_2} are all positive real numbers. We also consider that $t_1 < \mathrm{rvl}_{t_2} = \mathrm{cvl}_{t_2} = \mathrm{svl}_{t_2}$ and $r_1 < \mathrm{rvm}_{r_2} = \mathrm{cvm}_{r_2} = \mathrm{svm}_{r_2}$, i.e., a PR task consumes more resources compared to an NPR task, which complies with the first new characteristic of hybrid cloud rendering. In addition, according to the description of instances in Amazon EC2, a certain VM instance type (e.g., c4. xlarge for each PR VM) has the same VM size (4 vCPUs) and consumes the same optical bandwidth (750 Mbps). Thus we have $\mathrm{rvl}_{t_2} = \mathrm{cvl}_{t_2} = \mathrm{svl}_{t_2}$ and $\mathrm{rvm}_{r_2} = \mathrm{cvm}_{r_2} = \mathrm{svm}_{r_2}$. So, this is a very reasonable assumption.

3.2 Notation definitions

To facilitate further discussion, we list important notations below from two parts: parameters and variables. For each part, we list notations with an alphabetic order in Tables 1 and 2.

3.3 Mathematical formulation

As mentioned above, during the process of node-level mapping, we pack as many VMs as possible into a single server. During the process of link-level mapping, multiple VLs are mapped into the same wavelength

Table 1 Parameters: Part 1

cvl_{t_2}	The virtual optical bandwidth of the PR CVL connecting the PR CVM to be mapped into CDC;
cvm_{r_2}	The size of a PR CVM;
D	The set of DCs, and $\|D\| = 3$;
E	The set of fiber links;
N	The set of OXC nodes;
r_1	The size of an NPR VM;
r_2	The multi-dimension vector, and $r_2 = \{\mathrm{rvm}_{r_2}, \mathrm{cvm}_{r_2}, \mathrm{svm}_{r_2}\}$;
rvl_{t_2}	The virtual optical bandwidth of the PR RVL connecting the PR RVM to be mapped into RDC;
rvm_{r_2}	The size of a PR RVM;
SC	The server capacity;
SR_s^1	The set of NPR tasks, each of which has the user node s;
SR_s^2	The set of PR tasks, each of which has the user node s;
SR_1	The set of NPR tasks, and $\mathrm{SR}_1 = \bigcup_{s\in N, s\notin D} \mathrm{SR}_s^1$;
SR_2	The set of PR tasks, and $\mathrm{SR}_2 = \bigcup_{s\in N, s\notin D} \mathrm{SR}_s^2$;
svl_{t_2}	The virtual optical bandwidth of the PR SVL connecting the PR SVM to be mapped into SDC, and $\mathrm{rvl}_{t_2} = \mathrm{cvl}_{t_2} = \mathrm{svl}_{t_2}$;
svm_{r_2}	The size of a PR SVM, and $\mathrm{rvm}_{r_2} = \mathrm{cvm}_{r_2} = \mathrm{svm}_{r_2}$;
t_1	The virtual optical bandwidth required by an NPR VL;
t_2	The multi-dimension vector, and $t_2 = \{\mathrm{rvl}_{t_2}, \mathrm{cvl}_{t_2}, \mathrm{svl}_{t_2}\}$;
WC	The wavelength capacity.

Table 2 Variables

f_j	The index of a server in DC j;
j	The DC to complete rendering task;
mn	Originating and terminating ends of a fiber link;
P_1	The index of an NPR VL (used to identify VLs with the same wavelength and end nodes);
P_2	The index of a PR VL;
s	The user node, $\forall s \in N, s \notin D$;
V_1	The index of an NPR VM (used to identify VMs with the same server and end nodes);
V_2	The index of a PR VM;
w	The wavelength index;
$\Gamma_{s,j,mn}^{w,P_1}$	Boolean variable, which is equal to 1 if an NPR VL between end nodes s and j with index P_1 is using wavelength w on fiber link (m, n). Note that, the number of NPR VLs between end nodes s and j can be computed by Dijkstra-based anycast routing offline;
$\Gamma_{s,j,mn}^{w,P_2}$	Boolean variable, which is equal to 1 if a PR VL between end nodes s and j with index P_2 is using wavelength w on fiber link (m, n). The number of PR VLs between end nodes s and j is equal to $\|\mathrm{SR}_s^2\|$;
$\Gamma_{s,j}^{f_j,V_1}$	Boolean variable, which is equal to 1 if an NPR VM of user s with index V_1 is consolidated into server f_j in DC j. The number of NPR VMs between end nodes s and j can be computed by Dijkstra-based anycast routing offline;
$\Gamma_{s,j}^{f_j,V_2}$	Boolean variable, which is equal to 1 if a PR VM of user s with index V_2 is consolidated into server f_j in DC j. The number of PR VMs between end nodes s and j is equal to $\|\mathrm{SR}_s^2\|$.

of a single fiber link. Thus with the above-mentioned system model, we formulate our problem by the following objective function:

$$\text{Minimize} \quad \mathrm{MC}, \tag{1}$$

$$\mathrm{MC} \geqslant \mathrm{MC}_1 \cdot \mathrm{MC}_2, \tag{2}$$

$$\mathrm{MC}_1 = w \cdot \max(\Gamma_{s,j,mn}^{w,P_1}, \Gamma_{s,j,mn}^{w,P_2}), \ \ \forall s, j, mn, P_1, P_2, \tag{3}$$

$$\mathrm{MC}_2 = f_j \cdot \max(\Gamma_{s,j}^{f_j,V_1}, \Gamma_{s,j}^{f_j,V_2}), \ \ \forall s, V_1, V_2, j. \tag{4}$$

We try to minimize the mapping cost (i.e., the number of consumed servers and wavelengths). Eq. (3) obtains the maximal wavelength index MC_1 that represents the number of consumed wavelengths, and

$\max(\Gamma^{w,P_1}_{s,j,mn}, \Gamma^{w,P_2}_{s,j,mn})$ indicates that a wavelength will be assigned by an increasing index once it has been occupied by a VL. Similarly, we utilize (4) to obtain the maximal server index MC_2 that represents the number of consumed servers. Also, $\max(\Gamma^{f_j,V_1}_{s,j}, \Gamma^{f_j,V_2}_{s,j})$ indicates that a server will be assigned by an increasing index once it has carried a VM. So, the minimization of the mapping cost is equivalent to minimizing the maximal server and wavelength indices among all DCs and fiber links, which is shown in (1). Note that, we utilize (2) to obtain MC, where we consider the multiplication not the simple plus operation because there exists the interdependence not the conflict relationship between MC_1 and MC_2.

In order to formulate the problem, the above objective must satisfy a number of constraints.

1. Constraints of node-level mapping. When we try to perform the mapping between VM and server, we will satisfy the following constrains.

Eq. (5) indicates that the number of VMs in a server is constrained by one server capacity,

$$\forall j, f_j: \quad \left[\left(\sum_{s\in N, s\notin D}\sum_{V_1}\Gamma^{f_j,V_1}_{s,j}\right) + \left(\sum_{s\in N, s\notin D}\sum_{V_2}\Gamma^{f_j,V_2}_{s,j}\right)\right] \leqslant \mathrm{SC}. \tag{5}$$

Eq. (6) ensures that no VM can be divided,

$$\forall\ s, V_{1\mathrm{or}2}: \quad \sum_{f_j}\Gamma^{f_j,V_{1\mathrm{or}2}}_{s,j} \leqslant 1. \tag{6}$$

Here, $V_{1\mathrm{or}2}$ denotes V_1 or V_2.

2. Constraints of link-level mapping. When we try to perform the mapping between VL and fiber link(s), the following constraints must be satisfied.

Eq. (7) ensures that the total virtual optical bandwidth of VLs mapped into wavelength w on fiber link (m, n) does not exceed one wavelength capacity,

$$\forall w, mn: \sum_{s\in N, s\notin D}\sum_{j\in D}\left[\left(\sum_{P_1}\Gamma^{w,P_1}_{s,j,mn}\right) + \left(\sum_{P_2}\Gamma^{w,P_2}_{s,j,mn}\right)\right] \leqslant \mathrm{WC}. \tag{7}$$

Eqs. (8) and (9) ensure that the number of VLs using wavelength w between end nodes s and j cannot exceed the number of link-disjoint paths between s and j,

$$\forall s, j, w, P_{1\mathrm{or}2}: \sum_{(s,m)\in E}\Gamma^{w,P_{1\mathrm{or}2}}_{s,j,sm} \leqslant 1, \tag{8}$$

$$\forall s, j, w, P_{1\mathrm{or}2}: \sum_{(n,j)\in E}\Gamma^{w,P_{1\mathrm{or}2}}_{s,j,nj} \leqslant 1. \tag{9}$$

Here, $P_{1\mathrm{or}2}$ denotes P_1 or P_2.

Eq. (10) ensures wavelength continuity for all VLs,

$$\forall k \neq (s,j), \forall s, j, w, P_{1\mathrm{or}2}: \quad \sum_{(m,k)\in E, m\neq s}\Gamma^{w,P_{1\mathrm{or}2}}_{s,j,mk} = \sum_{(k,n)\in E, n\neq j}\Gamma^{w,P_{1\mathrm{or}2}}_{s,j,kn}. \tag{10}$$

Theorem 1. The above problem is NP-hard.

Proof. In (2), if MC_1 is skipped, the problem will be degenerated to the NP-hard knapsack problem1) without any constraints of link-level mapping. Similarly, our problem will be transformed into the NP-hard graph coloring problem2) without any constraints of node-level mapping, if MC_2 is skipped.

As mentioned above, we utilize (2) to obtain MC, where we consider the multiplication not the simple plus operation because there exists an interdependence between MC_1 and MC_2. It also means that our problem is not an absolutely linear process though the graph coloring sub-problem can be degraded as a linear processing if MC_2 is skipped, and vice visa. Therefore, we cannot directly use linear programming approach to obtain the optimal solution, but a relaxed bound can still be determined by us, in order to demonstrate the effectiveness of our heuristics.

1) http://en.wikipedia.org/wiki/Graph_coloring.
2) http://en.wikipedia.org/wiki/Knapsack_problem.

3.4 Bound analysis

Based on the problem formulation, we have the following theorems about lower bounds.

Theorem 2. In a DC, the least number of consumed servers is from $\lceil \frac{|\mathrm{SR}_2|}{\varsigma} \rceil$ to $(\lceil \frac{|\mathrm{SR}_1|}{\theta \cdot \varsigma} \rceil + \lceil \frac{|\mathrm{SR}_2|}{\varsigma} \rceil)$.

Proof. Considering $r_1 < \mathrm{rvm}_{r_2} = \mathrm{cvm}_{r_2} = \mathrm{svm}_{r_2} = \mathrm{vm}_{r_2}$ (we utilize the positive real number vm_{r_2} to represent the unified size owned by all kinds of PR VMs.), we let $r_1 = \frac{1}{\theta} \cdot \mathrm{vm}_{r_2} = \frac{1}{\theta} \cdot \frac{1}{\varsigma} \cdot \mathrm{SC}$ since a VM size is always smaller than one server capacity, and the size of each kind of PR VM is larger than that of an NPR VM according to the first new characteristic of hybrid cloud rendering. Here, θ and ς are both positive real numbers. For example, if $\mathrm{SC} = 32$, $\mathrm{vm}_{r_2} = 16$, and $r_1 = 2$, then $\theta = 8$ and $\varsigma = 2$.

Given $|\mathrm{SR}_2|$ PR tasks, we totally have $|\mathrm{SR}_2|$ PR RVMs, $|\mathrm{SR}_2|$ PR CVMs, and $|\mathrm{SR}_2|$ PR SVMs, because the virtual network of a PR task has one PR RVM, one PR CVM, and one PR SVM as shown in Figure 1(c). During the process of node-level mapping, for a PR task, each kind of PR VM should be consolidated into the server with the same attribute, for example, the PR RVM should be consolidated into the server within RDC. Thus if we assume that the link-level mapping is successful, the least number of servers consumed by PR RVMs is bounded by

$$\Phi^{\mathrm{sev}}_{\mathrm{PR\_RVM}} = \left\lceil \frac{|\mathrm{SR}_2| \cdot \frac{1}{\varsigma} \cdot \mathrm{SC}}{\mathrm{SC}} \right\rceil, \text{ if DC } j = \mathrm{RDC}. \tag{11}$$

The upper part of (11) is the total size of $|\mathrm{SR}_2|$ PR RVMs. And the $\lceil x \rceil$ returns the smallest integer no smaller than x. Also, the least number of servers consumed by PR CVMs is bounded by:

$$\Phi^{\mathrm{sev}}_{\mathrm{PR\_CVM}} = \left\lceil \frac{|\mathrm{SR}_2| \cdot \frac{1}{\varsigma} \cdot \mathrm{SC}}{\mathrm{SC}} \right\rceil, \text{ if DC } j = \mathrm{CDC}, \tag{12}$$

and the least number of servers consumed by PR SVMs is also bounded by

$$\Phi^{\mathrm{sev}}_{\mathrm{PR\_SVM}} = \left\lceil \frac{|\mathrm{SR}_2| \cdot \frac{1}{\varsigma} \cdot \mathrm{SC}}{\mathrm{SC}} \right\rceil, \text{ if DC } j = \mathrm{SDC}. \tag{13}$$

Given $|\mathrm{SR}_1|$ NPR tasks, the total number of NPR VMs is equal to $|\mathrm{SR}_1|$, because the virtual network of an NPR task has only one NPR VM. We consider that all NPR VMs will be consolidated into the same DC, e.g., we consolidate all NPR VMs into RDC. Thus if we assume that the link-level mapping is successful, the least number of servers consumed by NPR VMs is bounded by

$$\Phi^{\mathrm{sev}}_{\mathrm{NPR\_VM}} = \left\lceil \frac{|\mathrm{SR}_1| \cdot \frac{1}{\theta} \cdot \frac{1}{\varsigma} \cdot \mathrm{SC}}{\mathrm{SC}} \right\rceil, \text{ if DC } j = \mathrm{RDC}. \tag{14}$$

Therefore, during the process of node-level mapping, the least number of consumed servers is bounded by

$$\begin{cases} \Phi^{\mathrm{sev}}_{\mathrm{NPR\_VM}} + \Phi^{\mathrm{sev}}_{\mathrm{PR\_RVM}} = \left\lceil \frac{|\mathrm{SR}_1|}{\theta \cdot \varsigma} \right\rceil + \left\lceil \frac{|\mathrm{SR}_2|}{\varsigma} \right\rceil, \ j = \mathrm{RDC}, \\ \Phi^{\mathrm{sev}}_{\mathrm{PR\_CVM}} = \Phi^{\mathrm{sev}}_{\mathrm{PR\_SVM}} = \left\lceil \frac{|\mathrm{SR}_2|}{\varsigma} \right\rceil, \ j = \mathrm{CDC} \text{ or } \mathrm{SDC}. \end{cases} \tag{15}$$

Theorem 3. For a fiber link, the least number of consumed wavelengths is from

$$\sum_{s \in N, s \notin D, j = \mathrm{CDC \ or \ SDC}} \left\lceil \frac{|\mathrm{SR}^2_s|}{\beta \cdot n_{s,j}} \right\rceil$$

to

$$\sum_{s \in N, s \notin D, j = \mathrm{RDC}} \left(\left\lceil \frac{|\mathrm{SR}^1_s|}{\alpha \cdot \beta \cdot n_{s,j}} \right\rceil + \left\lceil \frac{|\mathrm{SR}^2_s|}{\beta \cdot n_{s,j}} \right\rceil \right).$$

Proof. Since $t_1 < \mathrm{rvl}_{t_2} = \mathrm{cvl}_{t_2} = \mathrm{svl}_{t_2} = \mathrm{vl}_{t_2}$ (we utilize the positive real number vl_{t_2} to represent the unified virtual optical bandwidth size owned by all kinds of PR VLs.), we let $t_1 = \frac{1}{\alpha} \cdot \mathrm{vl}_{t_2} = \frac{1}{\alpha} \cdot \frac{1}{\beta} \cdot \mathrm{WC}$ since the virtual optical bandwidth of a VL is always smaller than one wavelength capacity, and the virtual optical bandwidth of each kind of PR VL is larger than that of an NPR VL according to the first new characteristic of hybrid cloud rendering. Here, α and β are both positive real numbers.

Given $|\mathrm{SR}_s^2|$ PR tasks from the user node s, we totally have $|\mathrm{SR}_s^2|$ PR RVLs, $|\mathrm{SR}_s^2|$ PR CVLs, and $|\mathrm{SR}_s^2|$ PR SVLs, because the virtual network of a PR task has one PR RVL, one PR CVL, and one PR SVL as shown in Figure 1(c). $n_{s,j}$ records the number of link-disjoint paths between s and DC j, which means that we will have $n_{s,j}$ wavelengths between user node s and DC j. In other words, $\mathrm{WC} \cdot n_{s,j}$ is the total wavelength capacity we can use between user node s and DC j. During the process of link-level mapping, for a PR task, each kind of PR VL should arrive to the DC with the same attribute, for example, the PR RVL should arrive to RDC. Thus if the node-level mapping is successful, the least number of wavelengths consumed by PR RVLs is bounded by

$$\Psi^{\mathrm{wav}}_{\mathrm{PR\_RVL}} = \sum_{s\in N, s\notin D, j=\mathrm{RDC}} \left\lceil \frac{|\mathrm{SR}_s^2| \cdot \frac{1}{\beta} \cdot \mathrm{WC}}{\mathrm{WC} \cdot n_{s,j}} \right\rceil, \text{ if DC } j = \mathrm{RDC}. \tag{16}$$

The upper part of (16) is the total virtual optical bandwidth of $|\mathrm{SR}_s^2|$ PR RVLs. Similarly, the least number of wavelengths consumed by PR CVLs is bounded by

$$\Psi^{\mathrm{wav}}_{\mathrm{PR\_CVL}} = \sum_{s\in N, s\notin D, j=\mathrm{CDC}} \left\lceil \frac{|\mathrm{SR}_s^2| \cdot \frac{1}{\beta} \cdot \mathrm{WC}}{\mathrm{WC} \cdot n_{s,j}} \right\rceil, \text{ if DC } j = \mathrm{CDC}, \tag{17}$$

and the least number of wavelengths consumed by PR SVLs is also bounded by

$$\Psi^{\mathrm{wav}}_{\mathrm{PR\_SVL}} = \sum_{s\in N, s\notin D, j=\mathrm{SDC}} \left\lceil \frac{|\mathrm{SR}_s^2| \cdot \frac{1}{\beta} \cdot \mathrm{WC}}{\mathrm{WC} \cdot n_{s,j}} \right\rceil, \text{ if DC } j = \mathrm{SDC}. \tag{18}$$

Given $|\mathrm{SR}_s^1|$ NPR tasks from the user node s, the total number of NPR VLs between user node s and RDC is also $|\mathrm{SR}_s^1|$ if all NPR VLs arrive to RDC, because the virtual network of an NPR task has only one NPR VL. Thus if we assume that the node-level mapping is successful, the least number of wavelengths consumed by NPR VLs is bounded by

$$\Psi^{\mathrm{wav}}_{\mathrm{NPR\_VL}} = \sum_{s\in N, s\notin D, j=\mathrm{RDC}} \left\lceil \frac{|\mathrm{SR}_s^1| \cdot \frac{1}{\alpha} \cdot \frac{1}{\beta} \cdot \mathrm{WC}}{\mathrm{WC} \cdot n_{s,j}} \right\rceil, \text{ if DC } j = \mathrm{RDC}. \tag{19}$$

Therefore, during the process of link-level mapping, the least number of consumed wavelengths is bounded by

$$\begin{cases} \Psi^{\mathrm{wav}}_{\mathrm{NPR\_VL}} + \Psi^{\mathrm{wav}}_{\mathrm{PR\_RVL}} = \sum_{s\in N, s\notin D, j=\mathrm{RDC}} \left(\left\lceil \frac{|\mathrm{SR}_s^1|}{\alpha\cdot\beta\cdot n_{s,j}} \right\rceil + \left\lceil \frac{|\mathrm{SR}_s^2|}{\beta\cdot n_{s,j}} \right\rceil \right), \ j = \mathrm{RDC}, \\ \Psi^{\mathrm{wav}}_{\mathrm{PR\_CVL}} = \Psi^{\mathrm{wav}}_{\mathrm{PR\_SVL}} = \sum_{s\in N, s\notin D, j=\mathrm{CDC\ or\ SDC}} \left\lceil \frac{|\mathrm{SR}_s^2|}{\beta\cdot n_{s,j}} \right\rceil, \ j = \mathrm{CDC\ or\ SDC}. \end{cases} \tag{20}$$

4 Efficient heuristics for our framework

In the previous section, we have demonstrated the NP-completeness of our problem and analyzed lower bounds. Since the problem is NP-hard, in this section, we develop efficient heuristics to solve it. Firstly, we consider that the following parameters are given with an alphabetic order in Table 3.

As mentioned above, the hybrid cloud rendering has its unique characteristics: (1) a PR task consumes more resources compared with an NPR task; (2) an NPR task can be handled by any DC, while a PR task should be tackled by all three kinds of DCs (i.e., rendering, compressing, and storage DCs). Considering these two characteristics, we develop a Smaller Virtual Resource First (SVRF) heuristic. SVRF first tackles NPR tasks and then PR tasks follow. Another heuristic is called as Manycast Routing First (MRF), where PR tasks will be first tackled and each of them will be served by all three kinds of DCs, so that it is easy for us to consolidate the following NPR VMs into the server in any DC.

Table 3 Parameters: Part 2

D'	The set of candidate DCs	sr_1^i	The ith NPR task
M_j	The set of servers in DC j	sr_2^i	The ith PR task
$P_{s,j}^w$	The path between end nodes s and j with wavelength w	s_i	The ith user node
$\mathrm{RP}_{s,j}^w$	The free capacity of $P_{s,j}^w$		

Algorithm 1 for NPR tasks

Input: SR_1.
Output: MC_1, MC_2.

for $i = 1, 2, \ldots, |\mathrm{SR}_1|$ **do**
  $\forall \mathrm{sr}_1^i \in \mathrm{SR}_1$: $\langle s_i, t_1, r_1 \rangle$, execute FF strategy to establish the set $D' \leftarrow \mathrm{FF}(r_1, D)$;
  if $|D'| = 0$ **then**
    Block sr_1^i and the subsequent NPR tasks;
    Stop this algorithm.
  else
    while $|D'| \neq 0$ **do**
      $j = D'.top()$;
      Select **anycast routing** strategy;
      Execute Dijkstra to perform **link-level mapping** by using **traffic grooming**: $P_{s,j}^w \leftarrow \mathrm{Dijkstra}(s_i, j|\mathrm{RP}_{s,j}^w \geqslant t_1)$
      if $P_{s,j}^w$ can be found **then**
        $\mathrm{MC}_1 \leftarrow \arg_{\max}\{w\}$;
        Execute **node-level mapping** by using **server consolidation**: we select the first server f_j whose free capacity is not smaller than r_1 to accommodate this NPR VM;
        $\mathrm{MC}_2 \leftarrow \arg_{\max}\{f_j\}$;
        Break;
      end if
      $j = D'.pop()$;
    end while
    if $P_{s,j}^w$ cannot be found **then**
      Block sr_1^i;
    end if
  end if
end for
Return MC_1, MC_2.

4.1 Algorithm description

The pseudo code of serving NPR tasks is shown in Algorithm 1, and the pseudo code of handling PR tasks is shown in Algorithm 2. Obviously, SVRF executes Algorithm 1 and Algorithm 2 in order. MRF is a reverse process of SVRF. For the current rendering task, we first find the set D' of candidate destination DCs by using First Fit (FF) strategy. FF strategy is shown in Algorithm 3. We can see that from Algorithm 3, a candidate destination DC should have at least one server that has enough free capacity to accommodate the VM of the current rendering task. Followed by anycast routing, we block an NPR task only when $|D'| = 0$, because there does not exist any DC to complete this rendering task then. Similarly, followed by manycast routing, we will block a PR task if $|D'|<|D|$. No matter what kind of heuristic we utilize, traffic grooming and server consolidation will be used to achieve link- and node-level mappings, respectively.

After performing link- and node-level mappings in Algorithm 1, no matter what kind of DC, we will accept an NPR task, as long as we have one DC to complete this task. While in Algorithm 2, we accept a PR task until both link- and node-level mappings succeed for all three kinds of DCs. For each PR task sr_2^i, because $\mathrm{rvl}_{t_2} = \mathrm{cvl}_{t_2} = \mathrm{svl}_{t_2} = \mathrm{vl}_{t_2}$ and $\mathrm{rvm}_{r_2} = \mathrm{cvm}_{r_2} = \mathrm{svm}_{r_2} = \mathrm{vm}_{r2}$, we transfer $\langle s_i, D, t_2, r_2 \rangle$ into $\langle s_i, D, \mathrm{vl}_{t_2}, \mathrm{vm}_{r2} \rangle$.

4.2 Time complexity

The complexity of heuristics mainly depends on how many times we run FF strategy. Firstly, the complexity is bounded by $O(|M_j|\cdot|D|)$, if we run one time of FF strategy, which can be seen in Algorithm 3.

Algorithm 2 for PR tasks

Input: SR_2.
Output: MC_1, MC_2.
for $i = 1, 2, \ldots, |\mathrm{SR}_2|$ **do**
  $\forall \mathrm{sr}_2^i \in \mathrm{SR}_2$: $\langle s_i, D, \mathrm{vl}_{t_2}, \mathrm{vm}_{r_2} \rangle$, execute FF strategy to establish the set $D' \leftarrow \mathrm{FF}(\mathrm{vm}_{r_2}, D)$;
  if $|D'|<|D|$ **then**
    Block sr_2^i and the subsequent PR tasks;
    Stop this algorithm.
  else
    for $j = \mathrm{RDC}, \mathrm{CDC}, \mathrm{SDC}$ **do**
      Select **manycast routing** strategy
      Execute Dijkstra to perform **link-level mapping** by using **traffic grooming**:
      $P_{s,j}^w \leftarrow \mathrm{Dijkstra}(s_i, j|\mathrm{RP}_{s,j}^w \geqslant \mathrm{vl}_{t_2})$;
      if $P_{s,j}^w$ cannot be found **then**
        Block sr_2^i;
        Break;
      else
        $\mathrm{MC}_1 \leftarrow \arg_{\max}\{w\}$;
        Execute **node-level mapping** by using **server consolidation**: we select the first server f_j whose free capacity is not smaller than vm_{r_2} to accommodate this PR VM;
        $\mathrm{MC}_2 \leftarrow \arg_{\max}\{f_j\}$;
      end if
    end for
  end if
end for
Return MC_1, MC_2.

Algorithm 3 FF strategy (D' generation)

Input: (ω, D).
Output: D'.
$D' = \{\}$;
for $j = 1, 2, \ldots, |D|$ **do**
  for $f_j = 1, 2, \ldots, |M_j|$ **do**
    if the free capacity of server f_j is not smaller than ω **then**
      $D' \leftarrow D' + \{j\}$;
      Break;
    end if
  end for
end for
Return D'.

For Algorithm 1, we need to run FF strategy at most $(|\mathrm{SR}_1| \cdot |D'|)$ times, while for Algorithm 2, we need to run FF strategy at most $(|\mathrm{SR}_2| \cdot |D|)$ times. Therefore, the total complexity of heuristics is approximately $O[(|\mathrm{SR}_1| \cdot |D'| + |\mathrm{SR}_2| \cdot |D|) \cdot |M_j| \cdot |D|]$.

5 Simulation and analysis

In this section, we first introduce our simulation settings, and then discuss simulation results.

5.1 Simulation settings

In our simulations, we use NSFnet/RedIRIS as the substrate ODCN-level animation rendering infrastructure in Figure 2. RDC, CDC, and SDC connect the three largest-degree nodes. As for resource requirements, referring to the VM instances from Amazon EC2, we let $t_1 = r_1 = 2$ for each NPR task, and $\mathrm{vl}_{t_2} = \mathrm{vm}_{r_2} = 4$ for each PR task.

First of all, in order to demonstrate the reasonability of our bound analysis and the effectiveness of our heuristics, we consider **scenario 1**. In scenario 1, we let $\mathrm{SC} = \mathrm{WC} = 16$, i.e., we have $\theta = \alpha = 2$, $\varsigma = \beta = 4$. The number of NPR tasks $|\mathrm{SR}_1|$ increases from 100 to 800, and the number of PR tasks $|\mathrm{SR}_2|$ has the same variation range, i.e., $(|\mathrm{SR}_1|, |\mathrm{SR}_2|)$ increases from $(100, 100)$ to $(800, 800)$. According

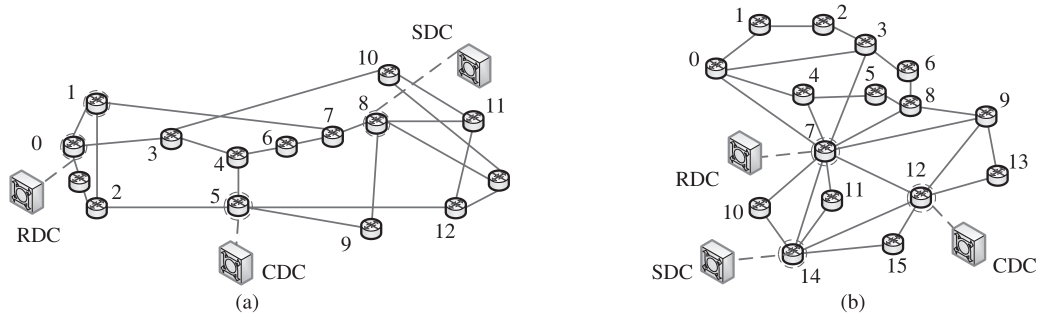


Figure 2 The ODCN-level animation rendering infrastructures. (a) NSFnet; (b) RedIRIS.

to Theorem 2 and Theorem 3, we can obtain the range of the lower bound for server cost and wavelength cost, respectively.

Next, we consider **scenario 2**, where the entire ODCN-level animation rendering infrastructure has limited resources. So, we cannot guarantee all rendering tasks can be served. In scenario 2, we let $(|\mathrm{SR}_1|, |\mathrm{SR}_2|) = (200, 200)$, $\mathrm{WC} = 16$ (i.e., $\alpha = 2$, $\beta = 4$), and $\mathrm{SC} = \{12, 16, 20, 24, 28\}$. Meanwhile, the number of servers is pre-determined as 40 for each DC, and the number of wavelengths is pre-determined as 10 for each fiber link.

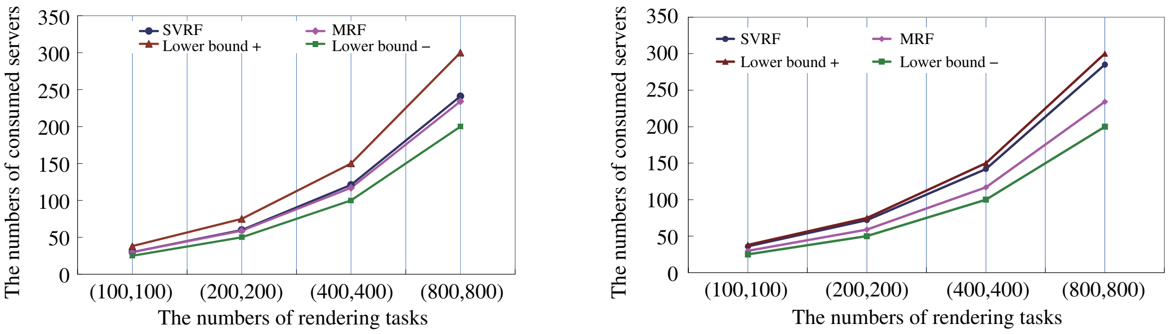
Under two scenarios, we run our simulations on a computer with an Intel Core i5 2.30 GHz CPU and 2 GB RAM.

It should be noted that due to the fact that this paper is the first work focusing on the ODCN-level virtual network embedding problem with the consideration of cloud rendering features, we compare the results of heuristics and theoretical lower bounds. Their good match will demonstrate the effectiveness of heuristics on reducing the mapping cost. Therefore, the following simulation results are very meaningful.

5.2 Simulation results

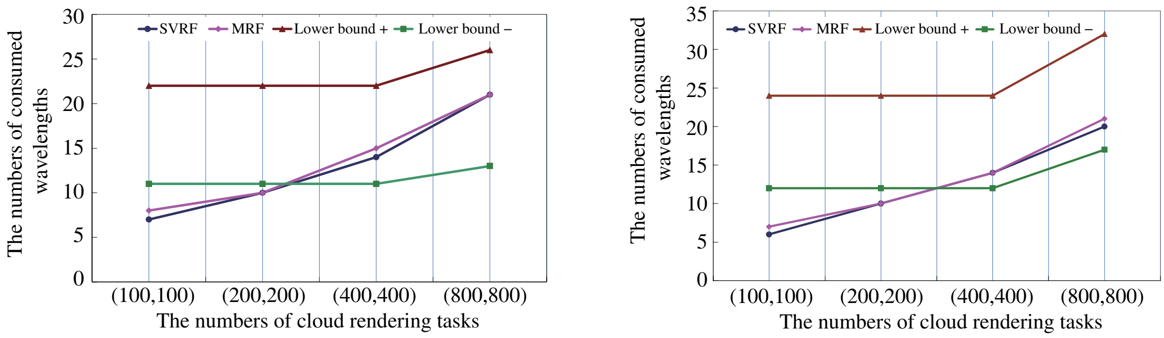
As shown in Figure 3(a), we use NSFnet as the substrate ODCN-level animation rendering infrastructure. And under scenario 1, according to Theorem 2, we obtain the range of the lower bound for server cost as follows: $[25, 38]$ at $(100, 100)$, $[50, 75]$ at $(200, 200)$, $[100, 150]$ at $(400, 400)$, and $[200, 300]$ at $(800, 800)$. Note that, '$[25, 38]$ at $(100, 100)$' indicates that the least number of consumed servers vary from 25 (lower bound $-$) to 38 (lower bound $+$) when $(|\mathrm{SR}_1|, |\mathrm{SR}_2|) = (100, 100)$. Similarly, under scenario 1, according to Theorem 3, we obtain the range of the lower bound for wavelength cost as follows: $[13, 26]$ at $(800, 800)$ and $[11, 22]$ at $(100, 100)$, $(200, 200)$, and $(400, 400)$. As shown in Figure 3(b), we use RedIRIS as the substrate animation rendering infrastructure. And we obtain the range of the lower bound for server cost as follows: $[25, 38]$ at $(100, 100)$, $[50, 75]$ at $(200, 200)$, $[100, 150]$ at $(400, 400)$, and $[200, 300]$ at $(800, 800)$. We also obtain the range of the lower bound for wavelength cost as follows: $[17, 32]$ at $(800, 800)$ and $[12, 24]$ at $(100, 100)$, $(200, 200)$, and $(400, 400)$.

In Figure 3(a), though it seems that we merely demonstrate the server cost not including wavelength cost, but for each fiber link, we let the number of wavelengths equal to the maximal lower bound of wavelength cost, i.e., 26 wavelengths per fiber link at $(800, 800)$ and 22 wavelengths per fiber link at $(100, 100)$, $(200, 200)$, and $(400, 400)$. Meanwhile, we vary the server cost so that we can determine the least number of consumed servers that ensures all rendering tasks can be served. As a result, the aforementioned process does not violate the integrated objective of minimizing server and wavelength costs mentioned in our problem formulation. Similarly, in Figure 3(b), we consider 32 wavelengths per fiber link at $(800, 800)$ and 24 wavelengths per fiber link at $(100, 100)$, $(200, 200)$, and $(400, 400)$. Meanwhile, we vary the server cost until we find the least number of consumed servers so that we serve all rendering tasks. From the simulation results in Figure 3 (a) and (b), we can see that the least number of consumed servers always well matches the bound range of server cost, whether SVRF or MRF. These results demonstrate the reasonability of our bound analysis and the effectiveness of our heuristics. In addition, the least server cost of two heuristics rises with the increasing number of rendering tasks. Finally, MRF performs better compared with SVRF, in terms of reducing server cost. This is because



(a) (b)

Figure 3 (Color online) Comparison of server cost among bound, SVRF and MRF. (a) NSFnet; (b) RedIRIS.



(a) (b)

Figure 4 (Color online) Comparison of wavelength cost among bound, SVRF and MRF. (a) NSFnet; (b) RedIRIS.

that MRF first tackles PR tasks, each of which will be served by all three kinds of DCs, so that it is easy for us to consolidate the following NPR VMs into the server in any one DC.

In Figure 4 (a) and (b), for each DC, we let the number of servers equal to the maximal lower bound of server cost, i.e., 38 servers at $(100, 100)$, 75 servers at $(200, 200)$, 150 servers at $(400, 400)$ and 300 servers at $(800, 800)$. Meanwhile, we vary the wavelength cost until we determine the least number of consumed wavelengths so that all rendering tasks can be served. As a result, the aforementioned process does not violate the integrated objective of minimizing server and wavelength costs mentioned in our problem formulation. From the simulation results in Figure 4 (a) and (b), we can see that the least number of consumed wavelengths well matches the bound range of wavelength cost, whether SVRF or MRF. It also demonstrates the reasonability of our bound analysis and the effectiveness of our heuristics. Moreover, the least wavelength cost of two heuristics increases when the number of rendering tasks follows a rising trend. Finally, SVRF performs slightly better compared with MRF, in terms of reducing wavelength cost. The reason for this is that SVRF first tackles NPR tasks, each of which will be served by any DC, so that it is easy for us to map the following PR VLs into a wavelength compared with MRF.

Under scenario 2, we demonstrate the number of blocked rendering tasks on NSFnet as shown in Figure 5(a) and on RedIRIS as shown in Figure 5(b), respectively. From the simulation results, we can see that the number of blocked rendering tasks decreases with the increment of server capacity, whether it is SVRF or MRF. More importantly, when the server capacity is not larger than 20, SVRF performs much better in terms of embedding rendering tasks; while once the server capacity becomes enough, such as $\mathrm{SC} = \{24, 28\}$, MRF performs slightly better. The reason for this is that if we serve PR tasks ahead followed by MRF, we can establish and select a large set of lightpaths and servers to hold or consolidate the following NPR tasks as long as the capacity of each wavelength/server is still enough. We can also count the running time of both heuristics on NSFnet as shown in Figure 6(a) and on RedIRIS as shown in Figure 6(b), respectively. We can observe that the running time varies between 4 and 10 s, which is acceptable.

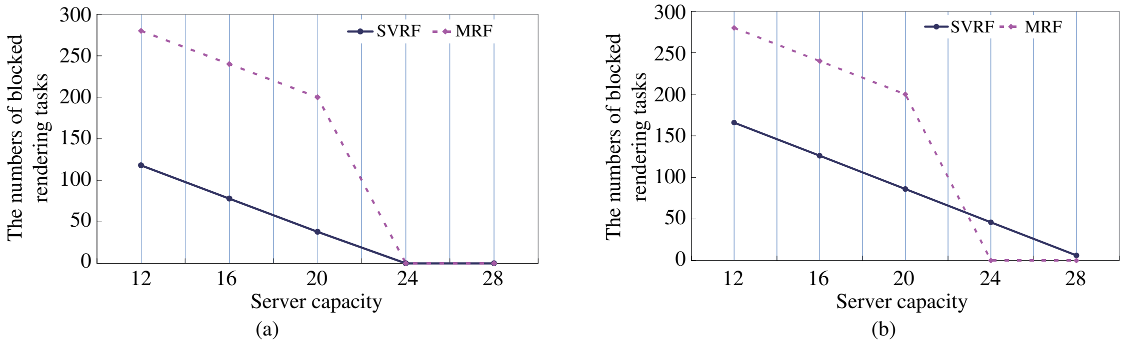


Figure 5 (Color online) Variation of the number of blocked rendering tasks with the increment of server capacity. (a) NSFnet; (b) RedIRIS.

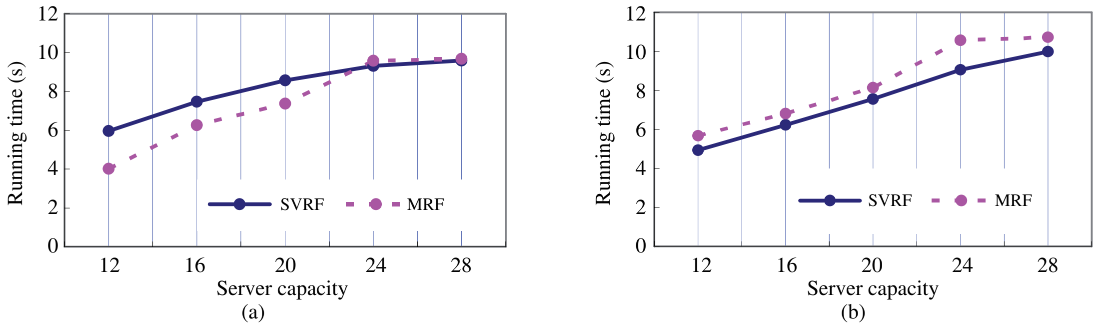


Figure 6 (Color online) Running time of two heuristics. (a) NSFnet; (b) RedIRIS.

6 Related work

In the ODCN with wavelength division multiplexing, the existing solutions of virtual network embedding mainly focused on the programmable task initiated by a certain user node, and they are mainly achieved by anycast and manycast routing principles. If a single DC can tackle a programmable task, anycast routing should be executed to establish the connection between user and the selected DC. But some programmable tasks must be completed by servers from different DCs, and manycast routing should be executed to build a tree-based connection from user to a list of specific DCs. In [4], a request of virtual network embedding is abstracted into the requirement of establishing a Virtual Lightpath (VL), and these requests are treated sequentially. For each request, the updated optical bandwidth and computing resources are the inputs of the integer linear programming that returns the result of the VL to be established. In [5,6], a mathematical model was first presented to reflect the linear energy growth of fiber links and servers. Through learning from variable information of energy consumption, the most energy-efficient VL can be determined. Considering the multi-priority requirement of establishing VLs, a multi-period virtual network embedding solution was proposed in [7]. Here, a high-priority requirement must be processed instantly, and a low-priority requirement can be served anytime within a maximal delay. With an accurate estimation of time- and priority-varying requirements, the least energy-consuming VL could be found at the current time period. The authors in [8] simultaneously generated multiple virtual networks. Each virtual network had a list of pre-established VLs, and it is unique to a specific user group, which was similar to a secure Virtual Private Network (VPN). These solutions improve energy efficiency with a subset of OXCs, while in a highly dynamical cloud computing environment, the frequent start-up operation is impracticable. Moreover, a convergence of optical and DCN is negligible. For this end, we have performed the ODCN-level virtualization under the scenario of power outage and evolving recovery [9], but this preliminary work cannot minimize the mapping cost during the process of virtual network embedding. And all existing solutions neglect the unique characteristic of hybrid cloud rendering.

7 Conclusion

To decrease the design cycle, cloud rendering was put forward and performed over the substrate ODCN. In the ODCN supporting cloud rendering, a rendering task will be tackled if we can embed its virtual network into the cloud renderfarm through node- and link-level mapping processes. Also, the implementation of hybrid cloud rendering should be considered. In this paper, considering the unique characteristic of hybrid cloud rendering, we have utilized traffic grooming and server consolidation to build a mapping-cost-aware cloud renderfarm from the problem formulation with lower bounds to heuristics. The simulation results have demonstrated the effectiveness of our methods in reducing the number of consumed servers and wavelengths (mapping cost). In the near future, we will focus on some real testbed implementations.

Acknowledgements
This work was supported in part by Open Foundation of State Key Laboratory of Networking and Switching Technology (Grant No. SKLNST-2013-1-08), Open Foundation of State Key Laboratory of Information Photonics and Optical Communications (Grant Nos. IPOC 2013B001, IPOC2014B009), Fundamental Research Funds for the Central Universities (Grant Nos. N140405005, N130817002, N130404002), Foundation of the Education Department of Liaoning Province (Grant No. L2014089), National Natural Science Foundation of China (Grant Nos. 61401082, 61471109, 61302070), Liaoning BaiQianWan Talents Program, and National High-Level Personnel Special Support Program for Youth Top-Notch Talent.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Assawamekin N, Kijsipongse E. Design and implementation of bit torrent file system for distributed animation rendering. In: Proceedings of IEEE International Computer Science and Engineering Conference, Nakorn Pathom, 2013. 68–72
2 Carroll D M, Hadzic I, Katsak A W. 3D rendering in the cloud. Bell Labs Tech J, 2012, 17: 55–66
3 Hu X Y, Sun B, Liang X H, et al. An improved cloud rendering method. In: Proceedings of IEEE International Conference on Image and Graphics, Xi'an, 2009. 853–858
4 Georgakilas K N, Tzanakaki A, Anastasopoulos M, et al. Converged optical network and data center virtual infrastructure planning. IEEE/OSA J Opt Commun Netw, 2012, 4: 681–691
5 Anastasopoulos M, Georgakilas K, Tzanakaki A. Evolutionary optimization for energy efficient service provisioning in IT and optical network infrastructures. In: Proceedings of IEEE European Conference and Exhibition on Optical Communication, Geneva, 2011. 1–3
6 Anastasopoulos M P, Tzanakaki A. Adaptive virtual infrastructure planning over interconnected IT and optical network resources using evolutionary game theory. In: Proceedings of IEEE International Conference on Optical Network Design and Modeling, Colchester, 2012. 1–5
7 Anastasopoulos M P, Tzanakaki A, Georgakilas K. Virtual infrastructure planning in elastic cloud deploying optical networking. In: Proceedings of IEEE International Conference on Cloud Computing Technology and Science, Athens, 2011. 685–689
8 Tzanakaki A, Anastasopoulos M P, Georgakilas K, et al. Energy aware planning of multiple virtual infrastructures over converged optical network and IT physical resources. In: Proceedings of IEEE European Conference and Exhibition on Optical Communication, Geneva, 2011. 1–3
9 Hou W G, Guo L, Liu Y J, et al. Virtual network planning for converged optical and data centers: ideas and challenges. IEEE Netw, 2013, 27: 52–58
10 Kantarci B, Mouftah H T. Energy-efficient cloud services over wavelength-routed optical transport networks. In: Proceedings of IEEE Global Telecommunications Conference, Houston, 2011. 1–5
11 Speitkamp B, Bichler M. A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Trans Serv Comput, 2010, 3: 266–278
12 Shen G, Tucker R S. Energy-minimized design for IP Over WDM networks. IEEE/OSA J Opt Commun Netw, 2009, 1: 176–186
13 Sun J, Wang Z W, Chen Z, et al. Parallel lightpath provisioning under wavelength continuity constraint in multi-domain optical networks based on hierarchical path computation element. Sci China Inf Sci, 2013, 56: 032114
14 Du S, Zhang S F, Peng Y F, et al. Power-efficient RWA in dynamic WDM optical networks considering different connection holding times. Sci China Inf Sci, 2013, 56: 042306