Hindawi Publishing Corporation
Scientific Programming
Volume 2016, Article ID 3976965, 14 pages
http://dx.doi.org/10.1155/2016/3976965

Research Article

VR-Cluster: Dynamic Migration for Resource Fragmentation Problem in Virtual Router Platform

Xianming Gao, Baosheng Wang, and Xiaozhe Zhang

College of Computer, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Xianming Gao; gxm9000@163.com

Received 27 December 2015; Revised 10 June 2016; Accepted 23 June 2016

Academic Editor: Ligang He

Network virtualization technology is regarded as one of gradual schemes to network architecture evolution. With the development of network functions virtualization, operators make lots of effort to achieve router virtualization by using general servers. In order to ensure high performance, virtual router platform usually adopts a cluster of general servers, which can be also regarded as a special cloud computing environment. However, due to frequent creation and deletion of router instances, it may generate lots of resource fragmentation to prevent platform from establishing new router instances. In order to solve "resource fragmentation problem," we firstly propose VR-Cluster, which introduces two extra function planes including switching plane and resource management plane. Switching plane is mainly used to support seamless migration of router instances without packet loss; resource management plane can dynamically move router instances from one server to another server by using VR-mapping algorithms. Besides, three VR-mapping algorithms including first-fit mapping algorithm, best-fit mapping algorithm, and worst-fit mapping algorithm are proposed based on VR-Cluster. At last, we establish VR-Cluster protosystem by using general X86 servers, evaluate its migration time, and further analyze advantages and disadvantages of our proposed VR-mapping algorithms to solve resource fragmentation problem.

1. Introduction

Network virtualization technology [1–3] has been regarded as a gradual solution to the development of Internet architecture, on which lots of researcher communities and equipment venders focus. In this context, service providers can rent network resources to lessees by establishing creating virtual networks. Meanwhile, they can recycle network resources occupied by virtual networks after the lifecycle of virtual networks comes to end. So, the future Internet can be regarded as "a big cloud computing environment," where operators can flexibly allocate and recycle network resources. For example, when one company wants to hold an online video conference, it can rent a virtual network for its business. In this way, users can get good quality of service during the lifecycle of virtual network. Besides, service provides can rent virtual network for research communities, and the latter can use these virtual networks to establish testbed and evaluate new network protocols in a real network environment.

Virtual router is one of key equipment to support the deployment of network virtualization technology, which is the same with traditional router in the Internet. Virtual router consists of several router instances, which can run in parallel and independently. And each router instance plays an important role in forwarding packets in virtual network. At the same time when service providers want to manage network infrastructure flexibly, network functions virtualization technology comes forth [4, 5]. The main idea behind it is to use general servers to establish network elements, such as virtual routers, firewall, NAT, and IDS. Besides, using general servers to establish packet processing devices catches more and more researchers' attentions, because the capability of packet processing in general server can be up to 40 Gbps by multicore and multithread technology. Under this environment, using a cluster of servers to achieve virtual router platform is a good solution [6]. So, service providers can flexibly create or release router instances in virtual router platform.

With service providers continually creating and deleting router instances, there is much discontiguous resource fragmentation in virtual router platform. Unfortunately, it cannot dispose of the incoming requests of creation of router instances any more, even though virtual router platform has enough resources. For instance, one platform consists of two servers with 10 Gbps processing ability: Server-A and Server-B. And service providers have established one router instance with 4 Gbps processing ability in Server-A and another router instance with 4 Gbps processing ability in Server-B. At the moment, it has two kinds of resource fragmentation on different servers: one resource block with 6 Gbps processing ability in Server-A and another resource block with 6 Gbps processing ability in Server-B. When a new request that is establishing router instance with 8 Gbps processing ability comes, it cannot create new router instance. Even though the remainder of resources in platform is up to 12 Gbps processing ability. In fact, resource fragmentation generated by frequent creation and deletion of router instances may prevent platform from responding to new requests when platform has enough resources. Thus, "resource fragmentation problem" is a severe problem. Unluckily, researchers do not pay any attention to the above problem, while putting lots of efforts into designing virtual router platform with high performance [7–11]. If we want to deploy virtual routers in the future, we should immediately start to attach importance to "resource fragmentation problem."

In this paper, we firstly put forward VR-Cluster platform to solve "resource fragmentation problem." It is established based on a cluster of servers by using the idea of network functions virtualization. From the aspect of function planes, it includes four function planes: switching plane, resource management plane, forwarding plane, and control plane. Forwarding plane and control plane also exist in the other virtual routers. Switching plane is a special plane, which uses flow strategies to transmit packets from exterior network to router instances and to transmit packets from router instances to exterior network. With the help of switching plane, operators can seamlessly migrate router instances from one server to another server without changing connection relationship between VR-Cluster platform and exterior network, which results in zero packet losses for migrated router instances. Resource management plane uses VR-mapping algorithm to dynamically calculate mapping relationship between router instances and physical platform and installs migration strategies into corresponding servers. And three VR-mapping algorithms including first-fit mapping algorithm, best-fit mapping algorithm, and worst-fit mapping algorithm are proposed, which are used to calculate mapping relationship between router instances and physical platform. At last, we establish a protosystem, namely, VR-Cluster, to support migration of router instances. VR-Cluster uses X86 servers to establish four function planes and uses switches to achieve full-mesh interconnection among function planes. Our experimental results show that VR-Cluster can migrate LF-Plane from one server to another server in one minute, which can meet requirements of dynamical migration. Besides, we also evaluate efficiency of our proposed VR-mapping algorithms based on resource model and evaluation model.

The remainder of this paper is organized as follow. Section 2 discusses and illustrates research work related to resource fragmentation in virtual router platform. Section 3 presents VR-Cluster architecture and exhibits how VR-Cluster supports dynamic migration. Section 4 establishes resource model for VR-Cluster and evaluation model for "resource fragmentation problem." Section 5 mainly exhibits three algorithms including first-fit mapping algorithm, best-fit mapping algorithm, and worst-fit mapping algorithm. Section 6 presents our experimental results including migration time and analysis of VR-mapping algorithms. Section 7 concludes this paper with a summary of our studies and discusses next works in the future.

2. Related Works

Many major router vendors and research communities have begun to follow suit in building support for router virtualization and explore how virtual routers can support multiple router instances running on the same underlying physical platform [2, 6–11]. It is a key device for bridging gap between network architectures and physical platforms. Thus, many people focus on moving toward virtual routers that can support polymorphic network architectures rather than monolithic network architectures and accommodate simultaneous coexistence of several router instances. However, unreasonable mapping relationship between router instances and virtual router platform may result in low resource utilization and increase the failure possibility of creation of router instances. We had explored it in our previous works [12].

Although "resource fragmentation problem" did not catch any researchers' attentions, some research communities had explored how router instances can move from one node to anther node [13–16]. For instances, RouterFarm [15] is proposed to support migration of router instances, which is achieved by reinstantiating a new router instance at a new location. However, operators need to reconfigure router instance, which may introduce downtime and packet loss. VROOM is another protosystem that is supporting for migration of router instances. In the earlier schemes of VROOM [16], which uses XEN to establish virtual environment and deploys router instance into each virtual environment, VROOM can move router instance from one location to another location by using XEN tools. In order to decrease downtime, current VROOM splits migration process into two steps: migration of control plane and migration of data plane. However, VROOM has to change network interfaces that is connecting to external network, which also results in changes of link status and recalculation of routing information.

However, existing schemes cannot achieve seamless migration of router instances, which may result in downtime and packet loss. In this paper, we try to use dynamical migration of router instances to solve "resource fragmentation problem." Thus, we should firstly put forward virtual router platform to support seamless migration before further analyzing VR-mapping algorithms.

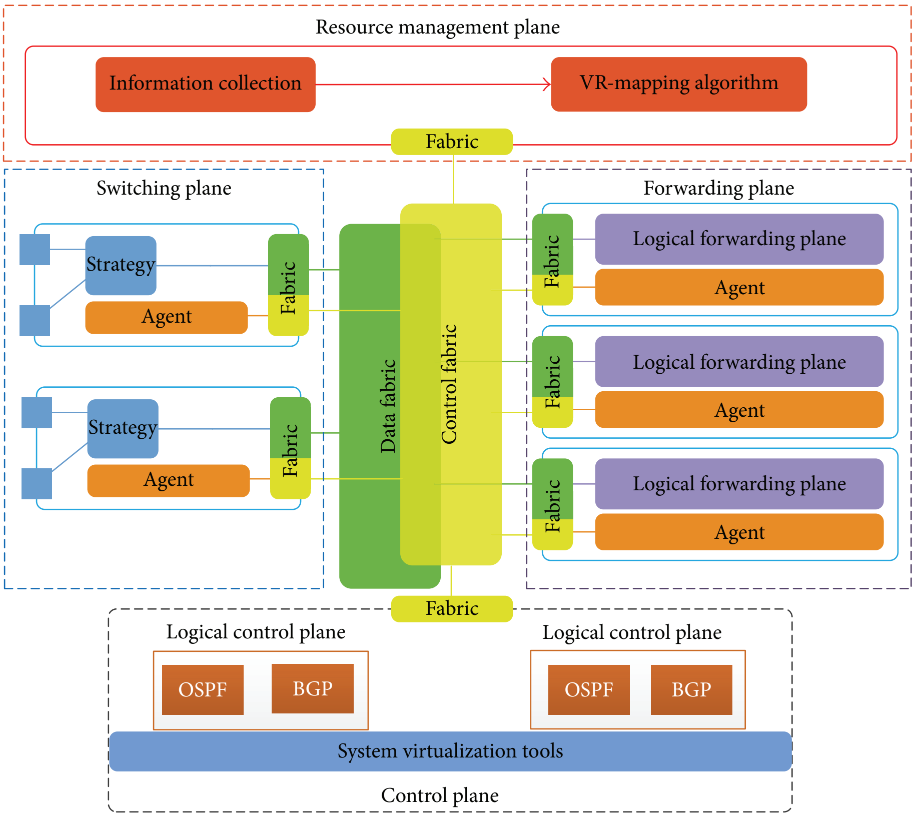


Figure 1: VR-Cluster platform architecture.

3. VR-Cluster

We mainly present VR-Cluster from two aspects: (1) platform architecture, which mainly includes four function planes, and (2) migration flow, which presents migration flow of router instance.

3.1. Platform Architecture. In order to support dynamic migration of router instances and to use dynamic migration to solve resource fragmentation problem, we put forward VR-Cluster architecture. Compared with other virtual router architectures, VR-Cluster includes four function planes: resource management plane, control plane, forwarding plane, and switching plane, as shown in Figure 1. Resource management plane and switching plane are customized function planes, which are used to support seamless migration of router instances.

VR-Cluster architecture mainly includes four layers: (1) resource management plane, which is responsible for managing platform, such as calculation of mapping relationship between router instances and physical platform, deployment of router instances, and release of occupied resources, (2) control plane, which usually uses system virtualization technology to run multiple logical control planes, (3) forwarding plane, which is used to run multiple logical forwarding planes, and (4) switching plane, which is responsible for connection between VR-Cluster and external network. VR-Cluster uses data fabric to achieve full-mesh interconnection between switching plane and forwarding plane. And control fabric is used to transmit control messages, such as configuration messages installed by resource management plane and control packets forwarded to control plane.

In VR-Cluster, forwarding plane and switching plane use "resource pool" idea, which consists of a cluster of servers. Forwarding plane can expand its forwarding ability by adding the quantity of servers; switching plane can flexibly change the amount and type of network interfaces. The high elasticity of VR-Cluster is brought by interconnection network including data fabric and control fabric. Thus, VR-Cluster not only can strongly increase management flexibility, but also has a good expansibility to support its ability.

3.1.1. Resource Management Plane. Resource management plane is responsible for management tasks in VR-Cluster. It has two main components: information collection (IC) and VR-mapping algorithm (VR-MA), as shown in Figure 2. And it usually runs in a single server, which can interconnect with other function planes via control fabric.

IC component is mainly responsible for collecting physical information and VR information from other three function planes. For physical information, it needs to know the quantity of servers in each function plane and the amount of each resource in servers. For VR information, it should know the amount of router instances, the occupied resources of router instances, and the location of router instances. IC component periodically collects VR-Cluster information and notifies VR-MA component with this information.

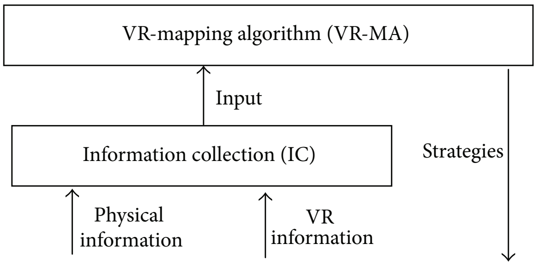


Figure 2: Resource management plane.

VR-MA component is used to calculate mapping relationship between router instances and physical platform based on collected information. After it completes calculation of migration schemes, it will send migration strategies to corresponding servers and execute migration actions. In this paper, we propose three VR-mapping algorithms and will present them later.

3.1.2. Control Plane. Control plane runs multiple logical control planes (LC-Planes for short) in parallel by using system virtualization technology, such as VMware [17], XEN [18], LXC [19], and KVM [20]. These LC-Planes share the same physical resources. And each LC-Plane can run either routing demos, such as Quagga [21] and XORP [22], or customized protocols to support new network architecture. Compared with forwarding plane and switching plane, control plane does not need to adopt "resource pool" idea, because a single server can support more than one hundred virtual environments in XEN or LXC. Although operators frequently create or recycle LC-Plane, it may not produce any resource fragmentation. Besides, the migration of logical control planes does not avoid downtime of router instances. Thus, VR-Cluster does not support migration of control plane. We think that migration of control plane just increases complexity of VR-Cluster, rather than resource utilization ratio.

Although VR-Cluster does not need to migrate control plane, it should cooperate with migration of forwarding plane. When one new logical forwarding plane is reinstantiated and sends registration messages to its corresponding LC-Plane, the latter will install routing tables and other configurations into this LF-Plane. And when the older LF-Plane is recycled by resource management plane, LC-Plane will not send any message to the older.

3.1.3. Forwarding Plane. Forwarding plane runs multiple logical forwarding planes (LF-Planes for short) in parallel. It does not use system virtualization technology to virtualize data plane, because using system virtualization to achieve forwarding plane has low forwarding ability. Now, each LF-Plane usually adopts function-based router (such as Click [23]) or flow-based router (such as OpenSwitch [24]). Each LF-Plane must have a corresponding LC-Plane in control plane.

Forwarding plane consists of a cluster of servers to meet various requests of forwarding ability in router instances. Resource fragmentation problem mainly results by creation and deletion of LF-Planes, rather than LC-Plane. At the same time, forwarding plane does not directly connect with exterior networks. When one LF-Plane is moved from one server to another server, it does not care about change of network interfaces. In other virtual router schemes, operators should adjust network interfaces that are connecting with exterior network while moving router instances, which does not avoid downtime of router instances. In VR-Cluster, forwarding plane and switching plane separation is a key design that can support seamless migration of forwarding plane.

When resource management plane installs migration strategies into forwarding plane, a new LF-Plane is established in forwarding plane. At the same time, LF-Plane should send registration messages to its corresponding LC-Plane and gets overall routing tables and other configurations from LC-Plane. Before new LF-Plane completes above action, the old LF-Plane is running in forwarding plane and is responsible for packet forwarding. Once resource management plane completes installation of routing tables, it configures switching plane, makes packets received from exterior network become sent to new LF-Plane, and recycles physical resources occupied by old LF-Plane.

3.1.4. Switching Plane. Switching plane is a special plane, which is used to provide network interfaces to connect with exterior network. Switching plane also adopts "resource pool" idea, as well as forwarding plane. Operators can expand the amount of network interfaces by adding servers in switching plane.

It adopts link virtualization technology to logically split physical network interface into several virtual network interfaces. Resource management plane establishes virtual network interfaces in switching plane and allocates these interfaces to corresponding router instances. Packets are continually received by virtual network interfaces in switching plane, and the latter sends packets to forwarding plane through data fabric based on flow strategies. Similarly, when forwarding plane forwards packets to exterior network, it firstly transmits packets to switching plane through data fabric and switching plane is responsible for packet transmission. Thus, operators can arbitrarily design their LF-Plane without caring about network interfaces that connected with exterior network.

Once these virtual interfaces are created, their lifecycle lasts until router instances are recycled by resource management plane, and resource management plane cannot move virtual network interface from one server to another server, as shown in Figure 3. If resource management plane changes network interface in one router instance, link status also changes, which can result in recalculation of routing tables. In some special scenes, it can breakdown end-to-end communication, which does not go against original idea in migration of router instances. Thus, VR-Cluster does not migrate network interface in router instances.

3.2. Migration Flow. Due to private features of virtual router, VR-Cluster just migrates LF-Plane, rather than the overall router instances, because it does not result in downtime of router instances and recalculation of routing tables. In this

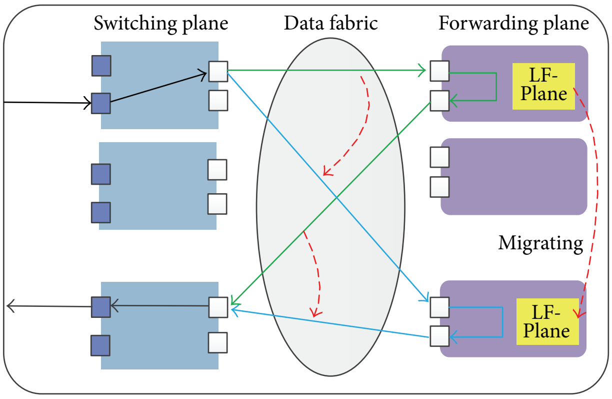


Figure 3: Interconnection relationship between switching plane and forwarding plane.

part, we mainly present how VR-Cluster can achieve seamless migration of router instances, as shown in Figure 4.

The procedure of migration of router instances in VR-Cluster mainly includes four steps, and tasks in each step are as shown in the following.

Step 1. There is a router instance in VR-Cluster. Packets received from exterior network are sent to LF-Plane through data fabric. If packets belong to control packets, they will be sent to LC-Plane by LF-Plane through control fabric. Besides, packets forwarded (or generated) by LF-Plane (or LC-Plane) must be sent to exterior network by switching plane. Resource management plane is responsible for checking whether or not there are migrating router instances, as shown in Figure 4(a).

Step 2. Resource management plane firstly establishes a new LF-Plane. Even though there are two LF-Planes, packets received by switching plane are just sent to old LF-Plane, rather than new LF-Plane. After resource management plane creates new LF-Plane, it immediately installs routing tables and configurations into new LF-Plane. Only when new LF-Plane has the same status as old LF-Plane, new LF-Plane has a chance to forward packets, as shown in Figure 4(b).

Step 3. Once when resource management plane has completed installation of routing tables and configurations, it will configure switching plane and control plane. And switching plane will send packets to new LF-Plane, instead of old LF-Plane. At the same time, it also configures control plane and makes the latter transmit control packets to new LF-Plane. However, resource management plane does not recycle old LF-Plane, because old LF-Plane also has some packets to not be disposed. Resource management plane does not recycle old LF-Plane until old LF-Plane has completed packet forwarding, as shown in Figure 4(c).

Step 4. Resource management plane recycles old LF-Plane. During migration flow of router instance, network interfaces do not change. LF-Plane may move from one server to another server. In this context, VR-Cluster completes seamless migration of router instance, as shown in Figure 4(d).

4. Mathematics Model

In this section, we firstly establish resource model for VR-Cluster and analyze main resources in each function plane. And then we establish evaluation model for resource fragmentation problem based on resource model in VR-Cluster.

4.1. Resource Model. We firstly analyze fundamental resource in VR-Cluster before we establish evaluation model for "resource fragmentation problem." VR-Cluster mainly includes three physical resources: control resource, forwarding resource, and switching plane. The three types of physical resources are responsible for different function planes. Besides, resource model should also include resources occupied by router instances. So resource model mainly includes four types of resource as follows.

Control Plane Resource. It refers to physical resources in control plane, which determines whether available resources meet requirements of LC-Plane. Cx refers to physical resources in control server that run LC-Planes. Each Cx includes three parts of resources: (1) CPU, which is responsible for calculation applications; (2) link-bandwidth, which is mainly used to communicate with its corresponding LF-Plane via control fabric; and (3) memory, which is used to store related information, such as link-state database, routing tables, and configuration rules, as shown in the following formula:

$$Cx = \sum \left(C_{\text{cpu}} + C_{\text{link}} + C_{\text{mem}} \right). \quad (1)$$

And C refers to the total physical resources in control plane, which is calculated as shown in the following formula:

$$C = \sum_{i=1}^{N} Ci \quad (i \in c). \quad (2)$$

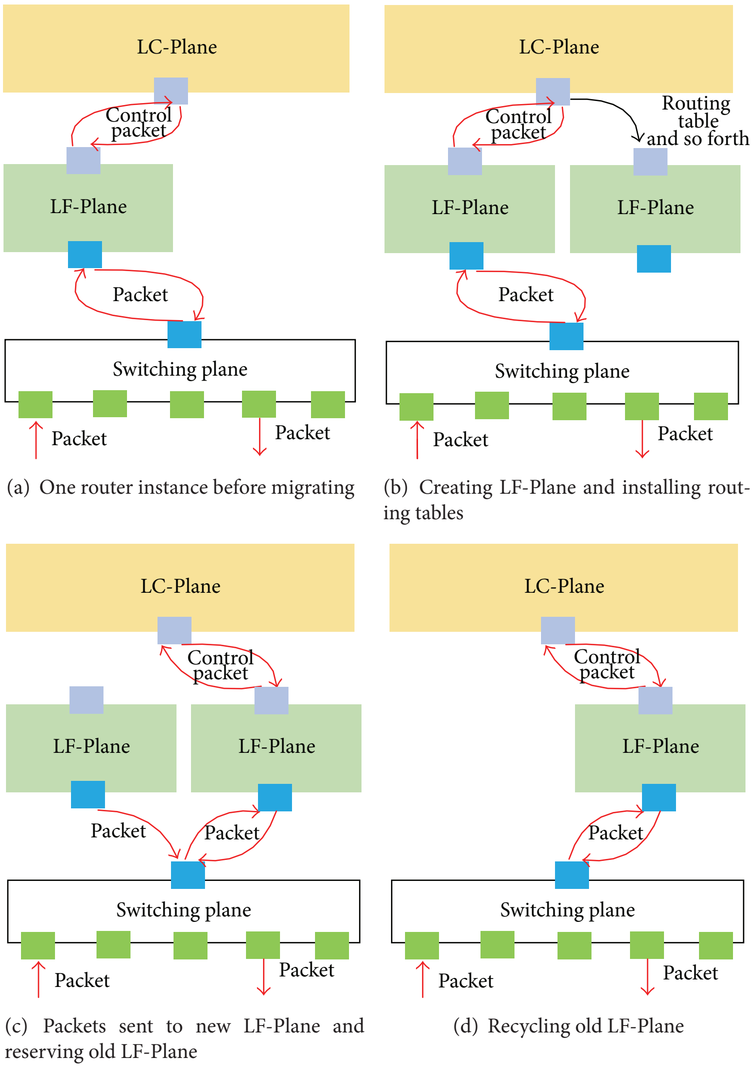
Forwarding Plane Resource. It refers to forwarding resources in forwarding plane. Fx refers to physical resource in forwarding server that runs LF-Planes. Each Fx also includes three parts of resources: (1) CPU, which is responsible for forwarding applications; (2) link-bandwidth, which is used to communicate with its corresponding LC-Plane via control fabric and to interconnect with switching plane via data fabric; and (3) memory, which is used to store some related information, such as forwarding tables or flow tables, as shown in the following formula:

$$Fx = \sum \left(F_{\text{cpu}} + F_{\text{link}} + F_{\text{mem}} \right). \quad (3)$$

And F refers to the total physical resources in forwarding plane, which is calculated as shown in the following formula:

$$F = \sum_{i=1}^{N} Fi \quad (i \in f). \quad (4)$$

Switching Plane Resource. It refers to network interfaces in switching plane. Sx refers to physical resource in switching



(a) One router instance before migrating

(b) Creating LF-Plane and installing routing tables

(c) Packets sent to new LF-Plane and reserving old LF-Plane

(d) Recycling old LF-Plane

FIGURE 4: Migration flow of VR-Cluster.

server that provides network interfaces interconnected with exterior network for router instances. Each Sx mainly consists of two types of network interfaces: (1) exterior interface, which is used to connect with exterior network, and (2) fabric interface, which interconnects with forwarding plane via data fabric, as shown in the following formula:

$$Sx = \sum \left(S_{\text{exterior}} + S_{\text{fabric}}\right). \quad (5)$$

And S refers to the total physical resources in switching plane, which is calculated as shown in the following formula:

$$S = \sum_{i=1}^{N} Si \quad (i \in f). \quad (6)$$

Router Instance Resource. It refers to physical resources occupied by router instances in VR-Cluster. Rx refers to physical resources occupied by one router instance, which includes three parts: (1) R_C^x, which stands for control resources occupied by LC-Plane in router instance x, (2) R_F^x, which stands for forwarding resources occupied by LF-Plane in router instance x, and (3) R_S^x, which refers to interfaces occupied by router instances x, as shown in the following formula:

$$Rx = R_C^x + R_F^x + R_S^x. \quad (7)$$

And R refers to the total physical resources occupied by router instances, which is calculated as shown in the following formula:

$$R = \sum_{i=1}^{N} Ri \quad (i \in VR). \quad (8)$$

We can use four types of resource to calculate resource utilization in VR-Cluster that is an important criterion. The resource utilization refers to the ratio of resources occupied by router instances to overall physical resources including overall control resources, overall forwarding resources, and overall interface-resources, as shown in the following formula:

$$R_{\text{utilization}} = \frac{R}{C + F + S}. \tag{9}$$

4.2. Evaluation Model. In order to evaluate "resource fragmentation problem" in VR-Cluster, we put forward three evaluation criterions: (1) failure magnitude of creation of router instances, (2) magnitude of resource fragmentation, and (3) ratio of resource fragmentation. The details and calculation formulas of these three evaluation criterions are presented as follows.

4.2.1. Failure Magnitude of Creation of Router Instances. This evaluation criterion is used to record failure times in the processing of creation of router instances. Resource management plane cannot establish new router instances, when it meets one of three conditions as shown in the following formula:

$$\text{sum}(\text{fail}) = \begin{cases} & \text{if } \left(\left(R_C^x > \max\{\operatorname{Re}m(Cx)\}\right) \&\& \left(R_C^x < C\right)\right) \\ \text{sum}(\text{fail}) + 1 & \text{if } \left(\left(R_F^x > \max\{\operatorname{Re}m(Fx)\}\right) \&\& \left(R_C^x < F\right)\right) \\ & \text{if } \left(\left(R_S^x > \max\{\operatorname{Re}m(Sx)\}\right) \&\& \left(R_C^x < S\right)\right) \\ \text{sum}(\text{fail}) & \text{other}. \end{cases} \tag{10}$$

In Formula (10), $\max\{\operatorname{Re}m(Cx)\}$ refers to the maximum of the remainder of resource block in control plane; $\max\{\operatorname{Re}m(Fx)\}$ refers to the maximum of remainder of resource block in forwarding plane; and $\max\{\operatorname{Re}m(Sx)\}$ refers to the maximum of remainder of interface in switching plane.

Formula (10) is mainly used to record failure magnitude of creation of router instances, when VR-Cluster has enough physical resources exceeding requested resources. If this evaluation criterion is too big, VR-Cluster has a severe resource fragmentation problem. Otherwise, operators can establish router instances with a low possibility of failure of creation of router instances. Thus, we can use Formula (10) to reflect resource fragmentation problem in VR-Cluster.

4.2.2. Magnitude of Resource Fragmentation. This evaluation criterion is used to refer to how much resource fragmentation is located in VR-Cluster. We can calculate the sum of resource fragmentation in each function plane, as shown in the following formula:

$$\text{sum}(\text{frag}) = \sum_{i}^{M}\sum_{p} P_{C(m)} + \sum_{j}^{N}\sum_{q} P_{F(n)} + \sum_{j}^{N}\sum_{q} P_{S(e)} \qquad (m \in C;\ n \in F;\ e \in S), \tag{11}$$

$P_{C(m)}$: it refers to resource fragmentation in control server m of control plane; $P_{F(n)}$: it refers to resource fragmentation in forwarding server n of forwarding plane; $P_{S(e)}$: it refers to resource fragmentation in interface server e of switching plane; $\sum_{i}^{M}\sum_{p} P_{C(m)}$: it refers to the magnitude of resource fragmentations in control plane; $\sum_{j}^{N}\sum_{q} P_{F(n)}$: it refers to the magnitude of resource fragmentations in forwarding plane $\sum_{j}^{N}\sum_{q} P_{S(e)}$: it refers to the magnitude of resource fragmentations in switching plane.

Magnitude of resource fragmentation is continually changing when operators frequently create and recycle router instances. If VR-Cluster has low magnitude of resource fragmentation, it has the higher chance to allocate physical resources for incoming requests of creation of new router instances; otherwise, it fails to establish new router instances, even though it has enough physical resources. Thus, magnitude of resource fragmentation is also an evaluation criterion to evaluate resource fragmentation problem in VR-Cluster.

4.2.3. Ratio of Resource Fragmentation. In order to further analyzing "resource fragmentation problem," we propose another evaluation criterion, namely, ratio of resource fragmentation. It refers to the ratio of the largest resource block to overall physical resource. The calculation of this evaluation criterion is as shown in the following formula:

$$f_R = \max\{f_C * \alpha, f_F * \beta, f_S * \chi\}. \tag{12}$$

In Formula (12), f_C refers to the ratio of resource fragmentation in control plane, as shown in Formula (13); f_F refers to ratio of resource fragmentation in forwarding plane, as shown in Formula (14); f_S refers to ratio of resource fragmentation in switching plane, as shown in Formula (15). In Formula (12), α, β, and γ are special parameters, whose value is just equal to 0 or 1.

When the ratio of resource fragmentation in control plane is not smaller than the ratio of resource fragmentation in forwarding plane and the ratio of resource fragmentation in switching plane, the ratio of resource fragmentation in VR-Cluster is determined by the ratio of resource fragmentation in control plane. And when both the ratio of resource fragmentation in control plane and the ratio of resource fragmentation in switching plane are smaller than the ratio of resource fragmentation in forwarding plane, the ratio of resource fragmentation in VR-Cluster is determined by the ratio of resource fragmentation in forwarding plane. f_R is always lower than 1; only when VR-Cluster does not establish any router instance, f_R is equal to 1.

Formula (13) is used to calculate the ratio of resource fragmentation in control plane: numerator refers to the largest resource block in control plane and denominator refers to the total resources in control plane:

$$f_C = 1 - \frac{\max\left(\text{size}\left(P_{C(n)}\right)\right)}{C}. \quad (13)$$

Formula (14) is used to calculate the ratio of resource fragmentation in forwarding plane: numerator refers to the largest resource block in forwarding plane and denominator refers to the total resources in forwarding plane:

$$f_F = 1 - \frac{\max\left(\text{size}\left(P_{F(n)}\right)\right)}{F}. \quad (14)$$

Formula (15) is used to calculate the ratio of resource fragmentation in switching plane: numerator refers to the largest resource block in switching plane and denominator refers to the total resources in switching plane:

$$f_R = 1 - \frac{\max\left(\text{size}\left(P_{S(e)}\right)\right)}{S}. \quad (15)$$

5. VR-Mapping Algorithm

VR-mapping algorithm mainly calculates mapping relationship between router instances and substrate infrastructure. Besides, it tries to solve "resource fragmentation problem" by adjusting mapping relationship.

We introduce three algorithms into VR-Cluster: first-fit mapping algorithm, best-fit mapping algorithm, and worst-fit mapping algorithm. So we hope that these three algorithms also have ability to get an optimization mapping relationship between router instances and physical infrastructure and calculate an optimization adjustment scheme, which can efficiently solve "resource fragmentation problem" in VR-Cluster. And we mainly consider mapping relationship between LF-Planes and forwarding plane, because only LF-Plane can migrate from one to another when resource management calculate migration strategy.

5.1. First-Fit Mapping Algorithm. In first-fit mapping algorithm, it mainly includes two types of linked lists: (1) F-idle list used to manage idle resource blocks in forwarding plane and (2) F-allocated list used to manage forwarding resources occupied by router instances.

In first-fit mapping algorithm, it selects migrated router instance from the last to the first in F-allocated list and searches the first resource block in F-idle link including enough remaining resources for the migrated router instance. Once one router instance is moved, first-fit mapping algorithm determines whether or not VR-Cluster can allocate resources for new router instance: if not, first-fit mapping algorithm will dynamically move router instance again until VR-Cluster can establish new router instance or there is no unmoved router instance. So, in first-fit mapping algorithm, router instances are always located in the front servers in forwarding plane.

The processing of first-fit mapping algorithm is present in Algorithm 1. When resource management plane migrates one router instance, it allocates forwarding resources for the latter. Firstly, first-fit algorithm searches F-idle list to find the first resource block that meets requirements of LF-Plane (S2). At last, if it finds its selected resource blocks in F-idle list, it can allocate resources for this request (S3–S5); otherwise, it should dynamically move LF-Plane in VR-Cluster. For dynamic migration in first-fit algorithm (S6–S16), it selects the last router instances (S7) and searches the first resource block for it (S8–S13): if there is one resource block for migrated LF-Plane, it will move LF-Plane to this resource block (S9–S12). If it does not find wanted result (S14–S17), it will continue searching the first resource block for the last but one until it searches the last router instance or gets its wanted result (S6).

5.2. Best-Fit Mapping Algorithm. In best-fit mapping algorithm, it mainly includes two types of linked lists: (1) F-idle list used to manage idle resource blocks in forwarding plane and (2) F-allocated list used to manage forwarding resources occupied by router instances.

In best-fit mapping algorithm, it selects migrated router instance from the last to the first in F-allocated list and searches the first resource block including enough remaining resources for migrated router instance in F-idle list. Once one router instance is moved, best-fit mapping algorithm determines whether or not VR-Cluster can allocate for new router instance: if not, best-fit mapping algorithm will dynamically move router instance again until VR-Cluster can establish new router instance or there is no unmoved router instance. Best-fit mapping algorithm, compared with first-fit mapping algorithm, has to sort resource blocks from smallest to largest after VR-Cluster moves router instances or establishes new router instance. So, in best-fit mapping algorithm, router instances may be located in some different servers in forwarding plane.

The processing of best-fit mapping algorithm is present in Algorithm 2. When resource management plane establishes one router instance, it allocates forwarding resources for the latter. Firstly, best-fit algorithm searches F-idle list to find the first resource block that meets requirements of LF-Plane (S2). At last, if it finds its selected resource blocks in F-idle list, it can allocate resources for this request and sort F-idle list from smallest to largest (S3–S7); otherwise, it should dynamically move router instances in VR-Cluster. For dynamic migration in best-fit algorithm (S8–S23), it selects the last LF-Plane (S9) and searches the first resource block for it (S10–S16): if there is one resource block for migrated LF-Plane, it will move LF-Plane to selected resource block and sort F-idle list from smallest to largest (S11–S15). If it does not search wanted result (S17–S22), it will continue searching the first resource block for the last but one until it searches the last LF-Plane or gets its wanted result (S8).

5.3. Worst-Fit Algorithm. In worst-fit mapping algorithm, it mainly includes two types of linked lists: (1) F-idle list used to manage idle resource blocks in forwarding plane and (2) F-allocated list used to manage forwarding resources occupied by router instances.

In worst-fit mapping algorithm, it selects migrated router instance from the last to the first in F-allocated list and

```
Input: migration of router instance
Output: whether or not execute it
Process:
    $N = \sum P_{F(n)}$; $R$ = amount of VR;
    while ($n < N$) {
        if ($P_{F(n)} \geq R_{F(n)}$)
            return SUCESS;
    }
    for ($i = R$; $i \leq R$; $i - -$){
        select $Ri$; //search the first resource block
        while ($n < N$){
            if ($P_{F(n)} \geq R^{i}_{F(n)}$) {
                move $Ri$ to $P_{F(n)}$;
                Break;
            }
        }    //determine whether or not VR-Cluster meets
        while ($n < N$){
            if ($P_{F(n)} \geq R^{x}_{F(n)}$)
                return SUCESS;
        }
    }
```

Algorithm 1: First-fit mapping algorithm.

```
Input: migration of router instance
Output: whether or not execute it
Process:
      $N \sum P_{F(n)}$; $R$ = amount of VR;
      while ($n < N$){
          if ($P_{F(n)} \geq R^{x}_{F(n)}$){
              sort F-idle list from smallest to largest;
          return SUCESS;
        }
      }
    for ($i = R$; $i \leq R$; $i - -$){
        select $Ri$; //search the first resource block
        while ($n < N$){
            if ($P_{F(n)} \geq R^{i}_{F(n)}$) {
                move $Ri$ to $P_{F(n)}$;
                sort F-idle list from smallest to largest;
                break;
            }
        }    //determine whether or not VR-Cluster meets
        while ($n < N$){
            if ($P_{F(n)} \geq R^{x}_{F(n)}$){
              sort F-idle list from smallest to largest;
              return SUCESS;
            }
        }
    }
```

Algorithm 2: Best-fit mapping algorithm.

searches the first resource block including enough remaining resources for migrated router instance in F-idle list. Once one router instance is moved, worst-fit mapping algorithm determines whether or not VR-Cluster can allocate new router instance: if not, best-fit mapping algorithm will dynamically move router instance again until VR-Cluster can establish new router instance or there is no unmoved router instance. Worst-fit mapping algorithm, compared with

```
Input: migration of router instance
Output: whether or not execute it
Process:
(1)   $N = \sum P_{F(n)}$; $R$ = amount of VR;
(2)   while ($n < N$){
(3)             if ($P_{F(n)} \geq R^{x}_{F(n)}$){
(4)                    sort F-idle list from largest to smallest;
(5)                    return SUCESS;
(6)             }
(7)   }
(8)   for ($i = R$; $i \leq R$; $i - -$){
(9)         select $Ri$; //search the first resource block
(10)        while ($n < N$){
(11)            if ($P_{F(n)} \geq R^{i}_{F(n)}$) {
(12)                 move $Ri$ to $P_{F(n)}$;
(13)                 sort F-idle list from largest to smallest;
(14)                 break;
(15)              }
(16)        }   //determine whether or not VR-Cluster meets
(17)        while ($n < N$){
(18)              if ($P_{F(n)} \geq R^{x}_{F(n)}$){
(19)                   sort F-idle list from largest to smallest;
(20)                   return SUCESS;
(21)              }
(22)        }
(23)  }
```

Algorithm 3: Worst-fit mapping algorithm.

first-fit mapping algorithm, has to sort resource blocks from largest to smallest after VR-Cluster moves router instances or establishes new router instance. So, in worst-fit mapping algorithm, router instances may be located in some different servers in forwarding resource pool and control resource pool.

The processing of worst-fit mapping algorithm is present in Algorithm 3. When resource management plane establishes one router instance, it allocates forwarding resources for the latter. Firstly, worst-fit algorithm searches F-idle list to find the first resource block that meets requirements of LF-Plane (S2). At last, if it finds its selected resource blocks both in F-idle-allocated list, it can allocate resources for this request and sort F-idle list from largest to smallest (S3–S7); otherwise, it should dynamically move LF-Plane in VR-Cluster. For dynamic migration in worst-fit algorithm (S8–S23), it selects the last LF-Plane (S9) and searches the first resource block for it (S10–S16): if there is one resource block for migrated LF-Plane, it will move LF-Plane to selected resource block and sort F-idle list from largest to smallest (S11–S15). If it does not search wanted result (S17–S22), it will continue searching the first resource block for the last but one until it searches the last LF-Plane or gets its wanted result (S8).

6. Experimental Results and Analysis

We firstly use general servers to achieve VR-Cluster including resource management plane, data plane, control plane, and switching plane. We then evaluate migration time based on VR-Cluster protoplatform. At last, we evaluate the amount of resource fragmentation, ratio of resource fragmentation, and failure magnitude of creation of router instances when VR-mapping algorithm uses different algorithms.

6.1. Experimental Environment. Our real implementation includes six servers and two switches, as shown in Figure 5: server with Intel Xeon CPU E5-2680 Processors, 64 GB RAM, dual 10 Gbps interfaces, and dual 10 Gbps interfaces, running Linux CentOS 6.5. One switch is 10 Gbps switch with 12 interfaces; another switch is 1 Gbps switch with 24 interfaces.

Control plane just consists of one server, which uses LXC to establish virtualization environment. The control interface interconnects with control fabric, so it can communicates with resource management plane and forwarding plane.

Resource management plane just includes one server. Its control interface interconnects with control fabric, which can communicate with control plane, forwarding plane, and switching plane.

Forwarding plane consists of two servers, running modified Click router. One 1 Gbps interface that is regarded as control interface interconnects with control fabric; one 10 Gbps interface that is regarded as data interface interconnects with data fabric.

Switching plane consists of two servers, running simple flow-table based application. One 1 Gbps interface that is regarded as control interface interconnects with control fabric; one 10 Gbps interface that is regarded as data interface interconnects with data fabric.

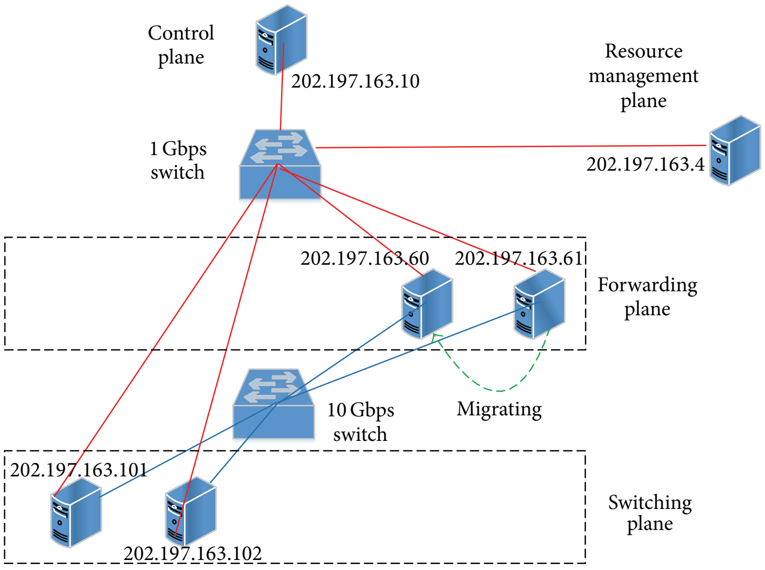


Figure 5: Experimental environment.

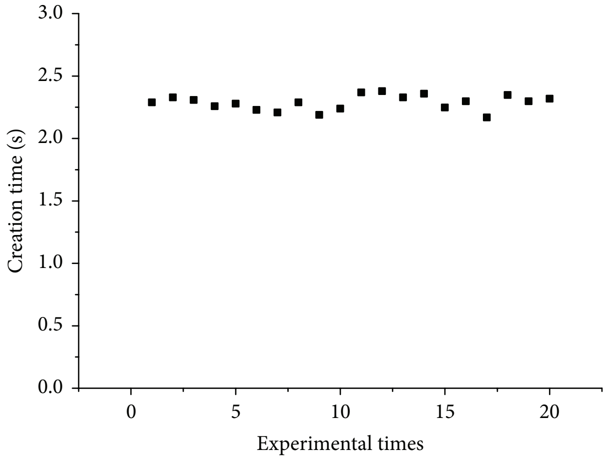


Figure 6: Creation time of LF-Plane.

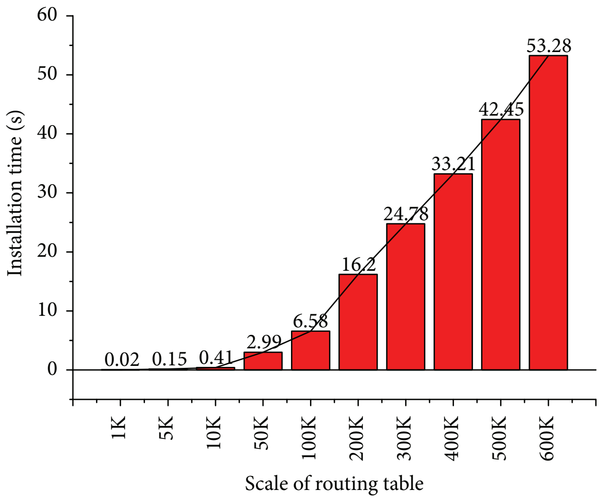


Figure 7: Installation time of routing table.

6.2. Migration Time. Migration time in VR-Cluster mainly includes two parts: (1) creation time of LF-Plane and (2) installation time of routing tables.

6.2.1. Creation Time of LF-Plane. In our VR-Cluster, LF-Plane adopts Click router to create LF-Plane. Besides, it uses DPDK tools to optimize I/O performance. When we migrate LF-Plane from one server to another server, we firstly create new LF-Plane in destination server. We evaluate twenty times of creation time of LF-Plane in forwarding plane, and our results are presented in Figure 6.

The average creation time of LF-Plane is about 2.3 s, because a new LF-Plane using DPDK tools has to take much time in initializing memory. And creation time is also affected by system loading in server of forwarding plane. If there are multiple LF-Planes, resource management plane uses more than 2.3 s to complete LF-Plane. Thus, creation time in VR-Cluster can meet our requirement, because old LF-Plane also continually forwards packets, which does not be beak off by creation of new LF-Plane.

6.2.2. Installation Time of Routing Table. In VR-Cluster, the time that LC-Plane takes to install a routing entry into its corresponding LF-Plane is about 0.3 ms. If the scale of routing table is 100 K, LC-Plane will takes about 55.4 s. In order to decrease installation time of routing table, we using packet batching technology to transmit multiple routing entries together. If MTU in control fabric is 1500, LC-Planes can install at most 121 routing entries once. We evaluate installation time of routing table under different scale of routing tables, as shown in Figure 7.

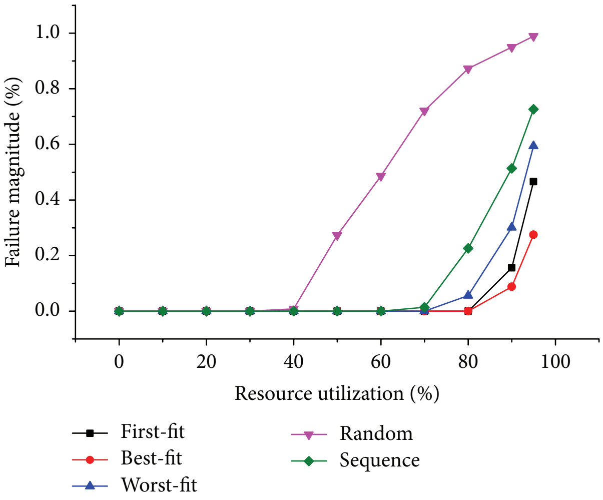


Figure 8: Failure magnitude.

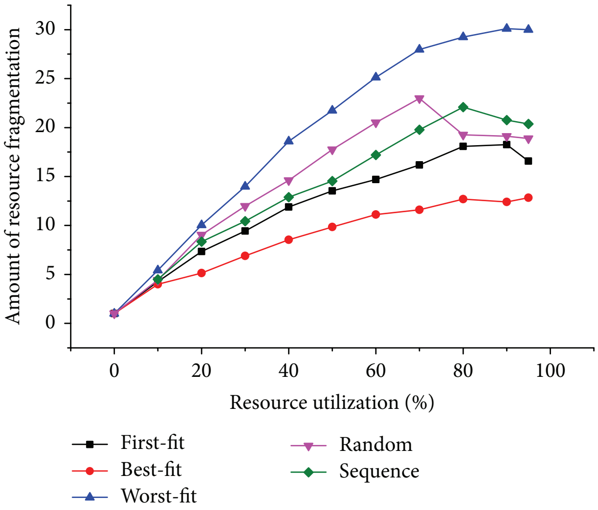


Figure 9: Amount of resource fragmentation.

In Figure 7, installation time continually grows with increase of scale of routing time. Relationship between scale of routing table and installation time does not exist linearly. Installation time it takes per time continually grows as the amount of routing entries it has installed increases, because LF-Plane has to take more time to search its routing tables when there is a huge amount of routing entries in it. For instance, LF-Plane takes about 0.56 ms to complete a single installation when it does not store any routing entry; when LF-Planes has stored 400 K routing entries, it may take about 1.09 ms.

6.2.3. Migration Time of LF-Plane. Migration time of LF-Plane is equal to the sum of creation time and installation time. When the scale of routing table is less than 10 K, migration time is mainly affected by creation time of LF-Plane. Once the scale of routing table is more than 50 K, migration time is mainly determined by creation time. In current Internet, a backbone router has about 500 K routing entries. In this context, migration time of LF-Plane is about 44.8 s, which is less than one minute. Thus, migration time of LF-Plane in VR-Cluster can meet our requirement.

6.3. Failure Magnitude of Creation of Router Instances. Failure magnitude of creation of router instances can be reduced by dynamic migration in VR-Cluster. In this part, we use it to evaluate whether or not dynamic migration is useful and to determine whether or not efficiency of dynamic migration is related to VR-mapping algorithms, as shown in Figure 8.

It shows that failure magnitude of creation of router instances increases as resource utilization continually increases. Random mapping algorithm firstly shows failure of creation of router instances when resource utilization is about 40%. And it has a worse performance as resource utilization continually increases. Sequence mapping algorithm is better than random mapping algorithm, because the latter may result in lots of small resource fragmentation. When resource utilization is too high, these two algorithms have a bad performance. For example, failure magnitude of creation of router instances in random mapping algorithm and sequence mapping algorithm is correspondingly 98.9% and 72.6% at 95% resource utilization. In our proposed three dynamic mapping algorithms, they show the first failure of creation of router instances only when the resource utilization is higher than 80%. Besides, performance of our proposed algorithms is better than the other two algorithms. In particular, best-fit mapping algorithm just has 27.5% failure magnitude when resource utilization is 95%. So, best-fit mapping algorithm is better than other proposed algorithms in terms of failure magnitude of creation of router instances.

6.4. Amount of Resource Fragmentation. Amount of resource fragmentation is another useful criterion to evaluate efficiency of dynamic migration. In this part, we mainly measure the amount of resource fragmentation generated in processing of creation and deletion of router instances, as shown in Figure 9.

Worst-fit mapping algorithm generates more resource fragmentation than other algorithms, because this algorithm usually allocates the largest resource fragmentation for router instances. Most of algorithms may generate more resource fragmentation as resource utilization continually increases. However, there is a special phenomenon in three algorithms including first-fit, random, and sequence mapping algorithm: the amount of resource fragmentation decreases after the resource utilization exceeds a defined value. Its reason is that VR-Cluster has a higher failure magnitude of creation of router instances (as shown in Figure 8) when resource utilization exceeds a defined value. However, the amount of resource fragmentation in best-fit mapping algorithm does not decrease, because it has a lower failure magnitude of creation of router instances. Best-fit mapping algorithm is also better than other algorithms, which always has no more than fifteen kinds of resource fragmentation.

6.5. Ratio of Resource Fragmentation. We measure the maximum size of resource fragmentation and calculate ratio of

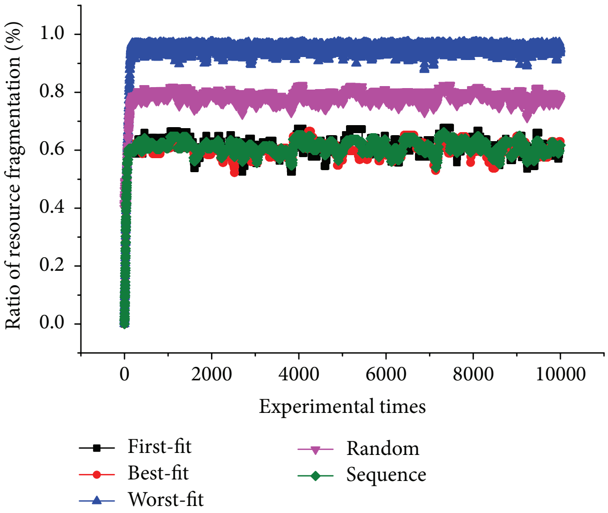


Figure 10: Amount of resource fragmentation.

resource fragmentation when resource utilization is 50%, as shown in Figure 10. When resource utilization is 50%, these algorithms except for random mapping algorithm do not show failure magnitude of creation of router instances.

It shows that ratio of resource fragmentation increases severely with a start and stays in a fixed position. The fixed value in worst-fit mapping algorithm is approximately 92%; the fixed value in random mapping algorithm is about 78%; and the fixed value of first-fit mapping algorithm is 60%, which is the same as the fixed value in best-fit and sequence mapping algorithm. The reason is that worst-fit mapping algorithm always allocates the maximal resources for router instances, and random mapping algorithm also potentially splits big resource fragmentation into small pieces of resource fragmentation. From aspect of ratio of resource fragmentation, first-fit algorithm and best-fit algorithm are better than worst-fit algorithm.

6.6. Summary. From the above experimental results, migration time in VR-Cluster can be accepted, which cannot result in downtime of router instances. Thus, VR-Cluster can use dynamic migration to solve "resource fragmentation problem."

We further explore efficiency of our proposed three VR-mapping algorithms. These three algorithms are better than random and sequence mapping algorithm. And best-fit mapping algorithm is the best in five mapping algorithms. It can reduce failure magnitude of creation of router instances, amount of resource fragmentation, and ratio of resource fragmentation.

7. Conclusion and Future Works

We find that there are lots of resource fragmentation in the processing of creation and deletion of router instances. And we firstly put forward "resource fragmentation problem" and establish evaluation model to analyze the above problem. In order to prove efficiency of dynamic migration, we establish a virtual router platform, called VR-Cluster, and use it to measure migration time of router instances. At the same time, this paper further proposes three VR-mapping algorithms including first-fit mapping algorithm, best-fit mapping algorithm, and worst-fit mapping algorithm. At last, experimental results prove that migration time of router instance can meet requirements of dynamical migration. Besides, best-fit mapping algorithm is the best to solve "resource fragmentation problem" in five algorithms.

In next works, we will explore a better VR-mapping algorithm than best-fit mapping algorithm, which can consider green energy, system loading balance, and so forth.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is supported by Program for National Basic Research Program of China (973 Program) "Reconfigurable Network Emulation Testbed for Basic Network Communication" and research on XXX access authentication and authorization protocol standards.

References

[1] N. M. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.

[2] D. Unnikrishnan, R. Vadlamani, Y. Liao et al., "Scalable network virtualization using FPGAs," in *Proceedings of the 18th ACM SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '10)*, pp. 219–228, Monterey, Calif, USA, February 2010.

[3] F.-E. Zaheer, J. Xiao, and R. Boutaba, "Multi-provider service negotiation and contracting in network virtualization," in *Proceedings of the 12th IEEE Network Operations and Management Symposium (NOMS '10)*, pp. 471–478, IEEE, Osaka, Japan, April 2010.

[4] D. Cotroneo, L. De Simone, A. K. Iannillo et al., "Network function virtualization: challenges and directions for reliability assurance," in *Proceedings of the 25th IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW '14)*, pp. 37–42, IEEE, Naples, Italy, November 2014.

[5] R. Mijumbi, J. Serrat, J. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, 2016.

[6] S. Bhatia, M. Motiwala, W. Muhlbauer et al., "Hosting virtual networks on commodity hardware," Georgia Tech Computer Science Technical Report GT-CS-07-10, 2008.

[7] J. S. Urner, P. Crowley, J. DeHart et al., "Supercharging planetlab: a high performance, multi-application, overlay network platform," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 85–96, 2007.

[8] M. B. Anwer and N. Feamster, "Building a fast, virtualized data plane with programmable hardware," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 75–82, 2010.

[9] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, "Fairness issues in software virtual routers," in

Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO '08), pp. 33–38, ACM, Seattle, Wash, USA, August 2008.

[10] G. Xie, P. He, H. Guan et al., "PEARL: a programmable virtual router platform," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 71–77, 2011.

[11] V. Eramo, E. Miucci, and M. Ammar, "Study of migration policies in energy-aware virtual router networks," *IEEE Communications Letters*, vol. 18, no. 11, pp. 1919–1922, 2014.

[12] X. Gao, B. Wang, X. Zhang et al., "Evaluation and analysis of three typical resource allocation algorithms in virtual router platform," in *Algorithms and Architectures for Parallel Processing: 15th International Conference, ICA3PP 2015, Zhangjiajie, China, November 18–20, 2015, Proceedings, Part IV*, vol. 9531 of *Lecture Notes in Computer Science*, pp. 549–568, Springer, Berlin, Germany, 2015.

[13] X. Chen and C. Phillips, "Virtual router migration and infrastructure sleeping for energy management of IP over WDM networks," in *Proceedings of the International Conference on Telecommunications and Multimedia (TEMU '12)*, pp. 31–36, Chania, Greece, August 2012.

[14] M. L. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[15] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: live router migration as a network-management primitive," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 231–242, 2008.

[16] Y. Wang, J. V. D. Merwe, and J. Rexford, "VROOM: virtual routers on the move," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 1–7, 2008.

[17] M. Dowty and J. Sugerman, "GPU virtualization on VMware's hosted I/O architecture," *ACM SIGOPS Operating Systems Review*, vol. 43, no. 3, pp. 73–82, 2009.

[18] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[19] D. Bernstein, "Containers and cloud: from LXC to docker to kubernetes," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, 2014.

[20] N. Pitropakis, D. Anastasopoulou, A. Pikrakis, and C. Lambrinoudakis, "If you want to know about a hunter, study his prey: detection of network based attacks on KVM based cloud environments," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–10, 2014.

[21] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador et al., "QuagFlow: partnering quagga with openflow," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 441–442, 2010.

[22] M. Handley, O. Hodson, and E. Kohler, "XORP: an open platform for network research," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 53–57, 2002.

[23] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.

[24] Y. Li and G. Wang, "SDN-based switch implementation on network processors," *Communications and Network*, vol. 5, no. 3, pp. 434–437, 2013.

Advances in Multimedia
The Scientific World Journal
International Journal of Distributed Sensor Networks
Journal of Industrial Engineering
Applied Computational Intelligence and Soft Computing
Advances in Fuzzy Systems
Modelling & Simulation in Engineering
Journal of Computer Networks and Communications
Advances in Artificial Intelligence
Advances in Computer Engineering
International Journal of Computer Games Technology
International Journal of Biomedical Imaging
Advances in Artificial Neural Systems
Advances in Software Engineering
Journal of Robotics
Advances in Human-Computer Interaction
Computational Intelligence and Neuroscience
International Journal of Reconfigurable Computing
Journal of Electrical and Computer Engineering