

## Research Article

# FSB-DReViSeR: Flow Splitting-Based Dynamic Replacement of Virtual Service Resources for Mobile Users in Virtual Heterogeneous Networks

Jean Frédéric Myoupo <sup>1</sup>, Vianney Kengne Tchendji <sup>2</sup>, Yannick Florian Yankam,<sup>2</sup>  
and Joël Casimir Tagne<sup>2</sup>

<sup>1</sup>Computer Science Lab-MIS, University of Picardie Jules Verne, 33 Rue St. Leu, 80039 Amiens, France

<sup>2</sup>Department of Mathematics and Computer Science, University of Dschang, Dschang, Cameroon

Correspondence should be addressed to Jean Frédéric Myoupo; jean-frederic.myoupo@u-picardie.fr

Received 11 May 2020; Revised 21 July 2020; Accepted 10 August 2020; Published 24 August 2020

Academic Editor: Liansheng Tan

Copyright © 2020 Jean Frédéric Myoupo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Virtual networks are sets of virtual devices that are interconnected through a physical network to provide services to end users. These services are usually heterogeneous (VOIP, VoD, streaming, etc.), exploit various amounts of resources (bandwidth, computing power, servers, etc.), and have topologies different from those of the substrate network. These variations in requirements are traditionally known as the architectural flexibility of virtual networks. Each virtual service is provided through a server called a virtual service resource. When a virtual service resource can no longer provide a good quality of service to end users due to the traffic variation generated by their mobility, two approaches are commonly implemented: provisioning the virtual network with resources or replacing the virtual service resource by migrating the service to another node that offers the most suitable amount of resource to satisfy the quality of service (QoS). In this paper, we propose a flow splitting-based dynamic virtual service resource replacement approach that allows for virtual service replacement across multiple virtual paths. Our approach is based on a graph topology that differs from those in the literature, which are based on tree topologies. The simulations performed in this study show that our approach significantly reduces the virtual service resource replacement time compared to other approaches.

## 1. Introduction

The great flexibility in network architecture offered to network operators (Cisco, Juniper, Amazon, etc.) by virtualization has increased interest in network virtualization for several years. With virtualization technology, a network service provider can create virtual networks using the resources of a physical network. Each of these virtual networks has its own topology; QoS and traffic management policies are ensured by the isolation property inherited from virtualization. Isolation allows the virtual equipment components to operate independently of each other, which enables moving from one physical environment to another or providing virtual resources to the users based on their needs

[1]. Today, many network technologies benefit from this possibility of free resource mobility. For example, cloud computing environments allow users to access on-demand heterogeneous resources (e.g., computing power, bandwidth, and storage). Even the widespread domain of the IoT (Internet of Things) is not excepted, with many heterogeneous pieces of equipment being used with large quantities of data ([2, 3]) produced every day. The IoT is a perfect illustration of a heterogeneous network paradigm that offers varied services with limited amounts of resources. In addition, when users move from one access point to another or when a node is added or removed, traffic is affected. The server that provides the service in this network (the virtual service resource) must consider this variation in traffic to

provide a good quality of service. However, the resources of this virtual service resource may become insufficient to meet this QoS requirement. To address this situation, additional resources can be provided to the server with a cloud computing infrastructure [2, 4]. In this case, the service provider must find a way to provide these resources while considering the number of users and profit constraints. This situation exemplifies the virtual network embedding (VNE) problem [5]. The solution to this problem has been widely explored in the literature ([5–7]), but all the proposed solutions reduce available physical resources. To prevent this decrease in resources, the virtual service resource can move from one source node to another with more available resources. This second approach does not require additional resources and has yet to be sufficiently explored [8, 9]; this study investigates this second approach. However, the proposed approaches implement a tree-based topology, while networks have a graph topology in practice.

*1.1. Related Works.* Regarding user’s mobility in the general context of wireless networks, the study in [10] proposed some resource allocation guidelines with maximization of the number of satisfied users. But these solutions just focus on the bandwidth and do not incorporate essential notions of QoS like jitter and packet loss rate. In addition, they do not occur in the context of virtual networks where the resource constraints are higher. In such environments, the actors (Service and Infrastructure providers) involved are numerous and diverse. In addition, the rapid evolution of the network size due to the flexibility of virtual networks and the freedom granted to each actor to design their network means that the resource allocation must also consider the architecture of these networks [11]. In optical networks, it will also be necessary to consider the communication spectrum [12]. But our study is not part of the optical networks.

In a more specific context of wireless networks, such as sensor networks or even IoT, the additional resources requirement is even greater. This is due to the fact that the objects (“things”) are generally not designed with large amounts of resources (storage and CPU). That is why the study in [13] proposed a cloud resources allocation model for the IoT information processing, in order to provide a satisfactory QoS to users, according to their traffic. The mathematical model proposed in [13] deals with the heterogeneity of user needs and IoT’s equipment in the allocation. But this solution emphasizes the search for the best allocation configuration and does not consider the latency times observed by users during the computations.

In examining the dynamic replacement of virtual service resources in the literature, Koyanagi and Tachibana [9] proposed a replacement method that satisfies QoS when a new node is added to the network. This method is based on a tree topology and consists of migrating the virtual service step by step to the new host node. At each intermediate node between the source and the destination, the method checks if the QoS is satisfied (Figure 1).

However, this approach has several limitations:

- (i) Replacing the service requires a lot of time, which does not meet the jitter (packet latency variation) requirement for good QoS, especially in virtual heterogeneous multimedia networks. This occurs because the QoS satisfaction is checked for each node in the replacement path.
- (ii) The tree topology used does not allow for the simultaneous exploration of other replacement possibilities.

Horiuchi and Tachibana [8] improved the approach of Koyanagi and Tachibana [9] by providing a replacement method that checks the QoS satisfaction only when the service has been fully replaced by the new host node. This significantly reduces the replacement time and jitter. However, the tree topology problem persists in this approach. In addition, the number of replacements is high, and no service migration method is discussed. The replacement algorithm of Horiuchi and Tachibana is shown in Figure 2, where node  $n_1$  is finally selected as the new virtual service resource.

To address the limitations of the approach in [8], we previously proposed a tree-based topology replacement approach that included a data migration technique [14]. To address the problem of equal traffic weight in the leaf nodes focused in [8], we proposed to explore the tree by going up until finding a level where the parent nodes have different traffic weights. The node with the highest traffic weight is then selected. The details of this method are provided in Figure 3. We recall here that, with tree leaves with identical traffic weights, Horiuchi and Tachibana [8] did not provide any solution.

In order to decrease the packet latency variation, we proposed in [14] to reduce the number of virtual service resource replacements. This helps to avoid unnecessary replacements. To complete that solution, we integrated an efficient data migration method. These solutions improved the virtual service resource replacement method proposed in [8]. However, the solutions proposed in [14] are still limited to a tree topology.

In the context of virtual machine migration, which is similar to the virtual service resource replacement studied in this paper, Liu et al. [15] recently proposed a virtual machine migration approach in an SDN-OpenFlow multicontroller environment. SDN (Software Defined Networking) has been shown to easily manage wide heterogeneous networks through a centralized control platform [16]. When migrating a virtual machine from one domain to another, the source machine asks the controller for the best migration path. Then, the controller selects the path with the highest bandwidth to route the migration packets. However, this approach has many drawbacks:

- (i) Incorrect routing decision in a graph including two paths with the same maximum bandwidth: let us consider the case of Figure 4. In this figure, we need to move a virtual machine from  $n_1$  to  $n_8$ . Three paths can be used: path 1 with 9 Gbps of bandwidth, path 2 with 8 Gbps of bandwidth, and path 3 with 9 Gbps of

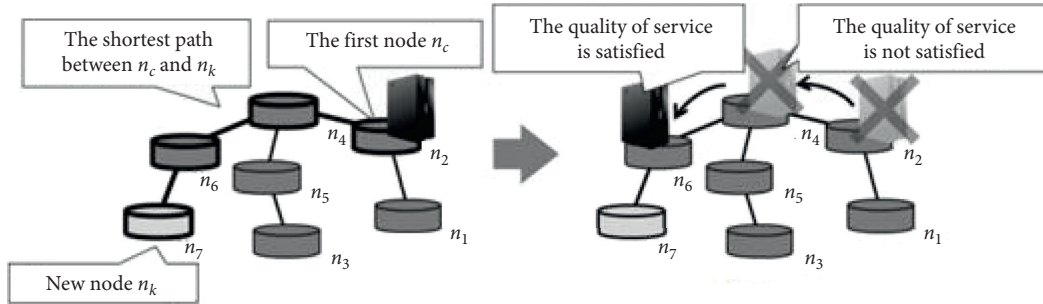


FIGURE 1: Virtual service resource replacement by moving the virtual network service by one hop.

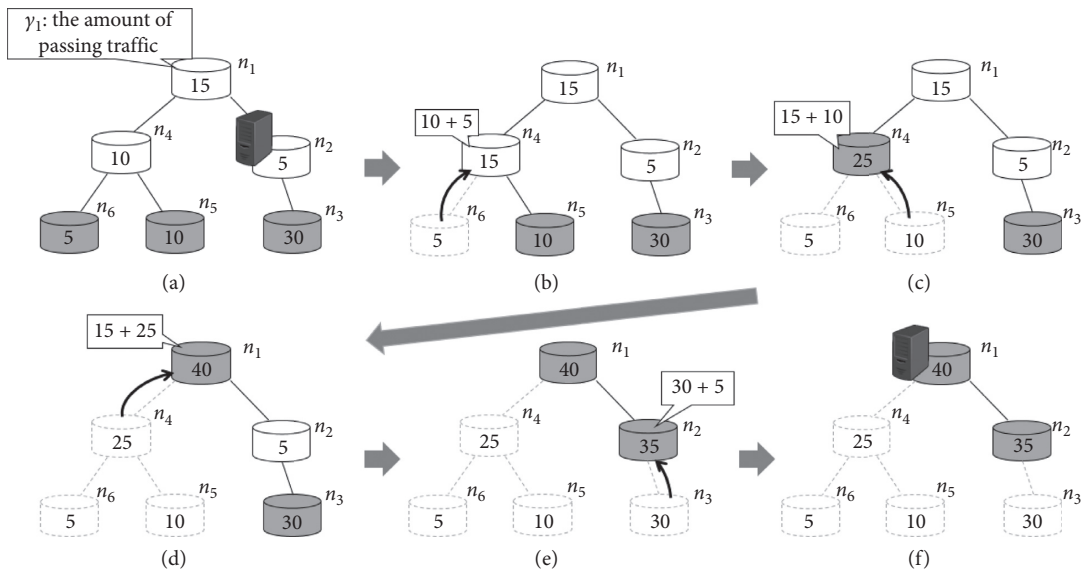


FIGURE 2: Single path virtual service resource replacement method of Horiuchi and Tachibana.

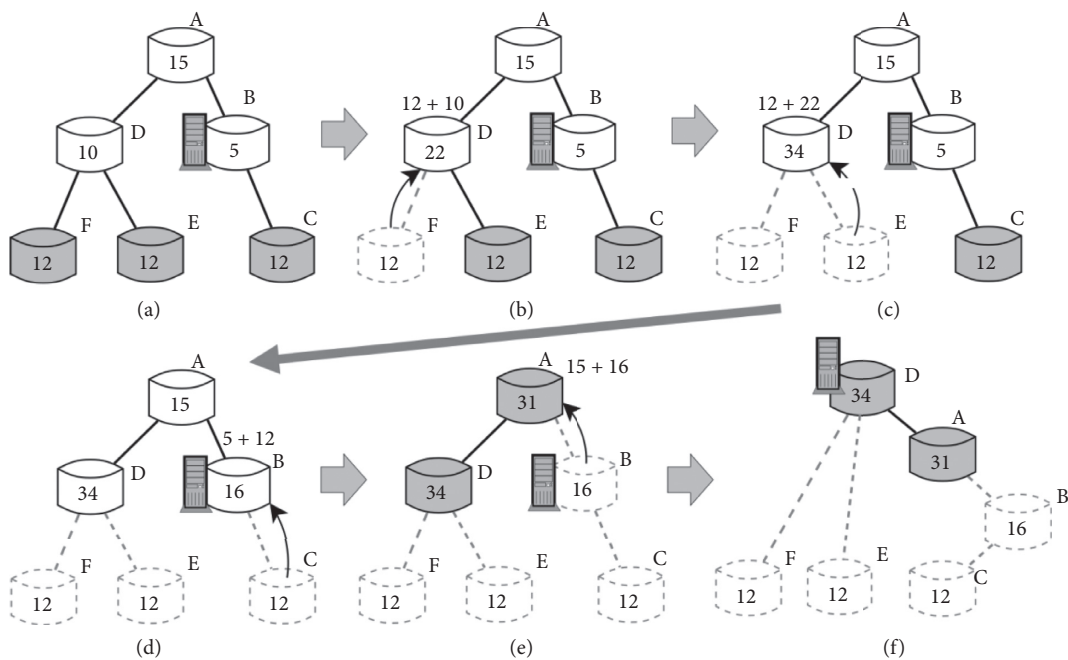


FIGURE 3: Solution to the equal traffic weight problem in the leaves nodes in [14].

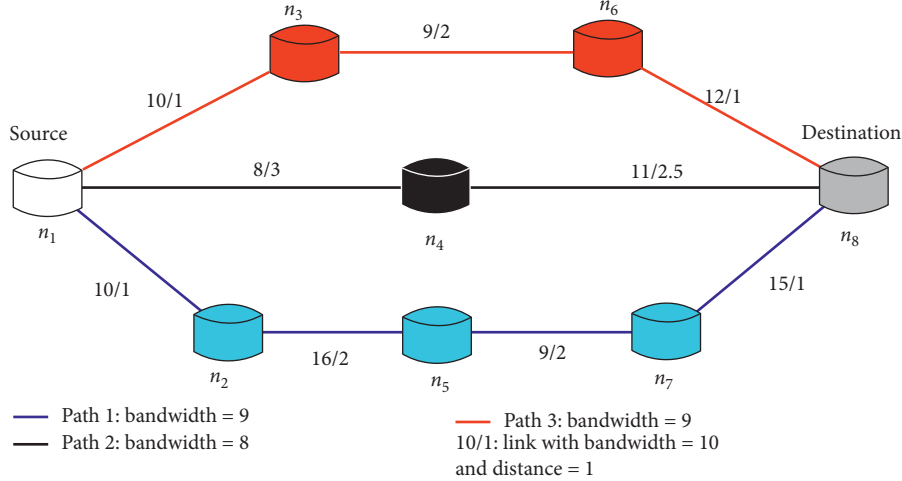


FIGURE 4: Incorrect routing decision in a case of multiple paths with the same bandwidth.

bandwidth. The method proposed by Liu et al. selects paths 1 or 3 as the most suitable migration path because they provide the greatest available bandwidth. Nevertheless, looking at the number of hops, path 3 is better than path 1.

- (ii) Invalid migration path decision in a graph with different bandwidth paths: considering the case of Figure 5, Liu et al. select path 1 as the most suitable for virtual machine migration because the bandwidths are  $5 > 4 > 3 > 2$ . However, considering the number of hops and the distance of the links along the different paths, we deduce the following results:
- Path1 (bandwidth = 5 Gbps): number of hops = 5 and overall distance =  $2 + 1 + 1 + 1 + 2.5 = 7.5$ .
  - Path2 (bandwidth = 2 Gbps): number of hops = 3 and overall distance =  $2.5 + 3 + 2.5 = 8$ .
  - Path3 (bandwidth = 3 Gbps): number of hops = 5 and overall distance =  $1 + 2 + 1 + 1 + 1 = 6$ .
  - Path4 (bandwidth = 4 Gbps): number of hops = 4 and overall distance =  $2 + 1 + 1 + 2 = 6$ .

Even though path 1 provides the greatest bandwidth, paths 2, 3, and 4 could be the best. Thus, when considered as a single criterion, bandwidth is not the most important criterion for a good migration path computation.

Once a path has been chosen, we can estimate the total migration time. According to Kherbache [17], the migration time of a virtual machine is provided by a linear model (1), where  $d_{\text{mig}}$  is the migration time and evolves linearly compared to the forwarded amount of memory  $m_{\text{mig}}$  and the bandwidth  $b_p$  available for migration. The interruption time  $i_{\text{mig}}$  describes how quickly the files of the virtual machine copy and is generally long but has very little impact on the migration time [17]. However, (1) does not consider the distance between the links or the number of crossed nodes:

$$d_{\text{mig}} = \frac{m_{\text{mig}}}{b_p} + i_{\text{mig}}. \quad (1)$$

*1.2. Our Contributions.* The main motivation of this paper is to improve QoS in heterogeneous virtual networks when traffic varies due to user mobility. Therefore, we provide solutions to the limitations noted in Horiuchi and Tachibana's work, especially those related to tree topology. Our contributions are summarized in two points:

- A multipath flow splitting replacement approach for a fast virtual service resource replacement: service files are migrated through several links to a destination node. This approach is based on the traffic splitting strategy used in the literature to manage cases of node and link failures ([18, 19]). The bandwidth, number of hops, and link distances are used as criteria to select the best path.
- We enhance our method by considering the throughput as another criterion to select the best path instead of the bandwidth. The bandwidth does not consider the real available flow space in the links, allowing us to select a path as the best replacement path, while the space available in the link is not sufficient to carry the data flow.

The remainder of this paper is organized as follows. In Section 2, we describe the traffic splitting method in detail. Section 3 presents our first contribution related to flow splitting-based replacement using the bandwidth, number of hops, and link distances as criteria to select the best path. Section 4 enhances our contribution to a flow splitting-based replacement approach using link throughput as a criterion for the best path selection criterion. Simulation results and discussions are presented in Section 5. Finally, Section 6 concludes the paper.

## 2. Motivation for Traffic Splitting

This section describes the traffic splitting approach and its main requirements.

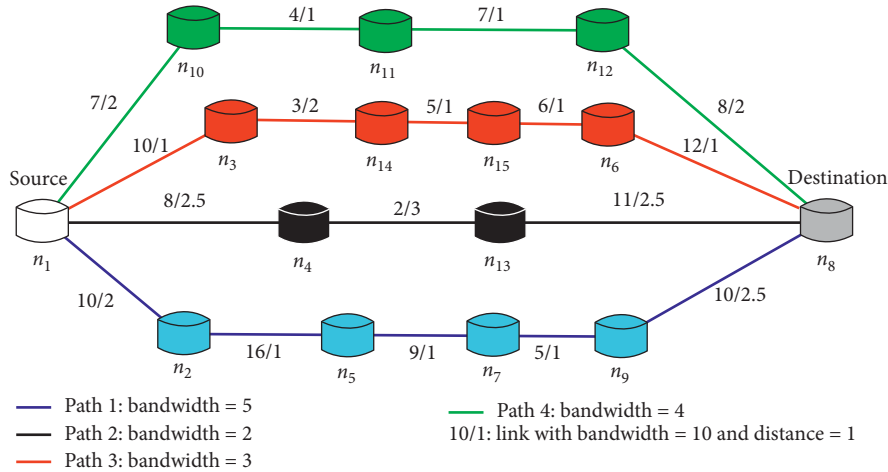


FIGURE 5: Graph with different bandwidth paths.

**2.1. Description.** Traffic splitting is a method that consists of forwarding a flow weight over several links to ensure end-to-end QoS [18, 19]. Traffic splitting is generally used for the following reasons:

- (i) Resource gap over links and nodes [18]: in this case, the load of the original flow is distributed over several links to avoid congestion. The gap is often caused by link or nodes failures, which involve the packets rerouting onto alternative links [18–20] and congestion due to user mobility and topology change [8, 9, 21].
- (ii) Exploiting idle resources in the network, such as mirror servers, to quickly process the requests of a large number of users: the flowlets (small flow coming from splitting) can be of equal weight or of different weight [22] depending on the available resources on the links.

In these cases, flows obtain more resources for rerouting than in a single path approach.

**2.2. Traffic Splitting Requirements.** One of the most important challenges when dealing with the flow splitting method is the packets reassembling at the destination node when the splitting motivation is rerouting [18]. Due to the heterogeneity of the links, packets do not always arrive at the destination in the transmission order, which yields difficulties in reconstructing the original flow. A solution to this problem in the literature is to minimize the number of flowlets [22] and to label the packets of the same flow so that they can be reassembled easily without loss ([18, 22]).

Another major challenge is to choose the best forwarding paths for the flowlets. The number of these paths determines the number of flowlets (subflows) that will be generated. This challenge deals with the criteria used to select the best paths for flow forwarding. Several selection criteria can be considered:

- (i) Link length: in this case, we choose paths that offer the minimum distance between the source and the

destination nodes for flowlet forwarding. However, we can have a minimal distance path that offers a low bandwidth but also a long forwarding time. This is not acceptable to ensure QoS.

- (ii) Bandwidth.
- (iii) Throughput.

The virtual service resource replacement method used in this paper uses these three criteria in the replacement best path selection process.

### 3. Our Virtual Service Resource Replacement Method with Traffic Splitting

We describe our replacement method based on traffic splitting, which overcomes the limits of the Horiuchi and Tachibana [8] approach. This method transcends the approach in [8] by dealing with graph topologies and data migration policies of virtual network services.

**3.1. Hypothesis and Notations.** Before describing our method, let us assume the following:

- (i) The network is represented by a graph  $G(N, L)$ , where  $N$  represents the set of nodes and  $L$  is the set of links.
- (ii) Links are bidirectional, and there are at least two disjointed paths between each pair of links, making it possible to connect them.
- (iii) Heterogeneous networks are supervised through an SDN controller.
- (iv) Each link is associated with a weight that represents the bandwidth, length, or throughput on that link, and the weight of the nodes is not considered.
- (v) The service copy interruption time is the same for all nodes.

Let us consider the following notations:

- (i)  $n_i$ : node number  $i$ .

- (ii)  $P_i$ : path number  $i$ .
- (iii)  $l_{ik}$ : link between the nodes  $n_i$  and  $n_k$ .
- (iv)  $d_{ik}$ : migration time from the node  $n_i$  to  $n_k$ .
- (v)  $bp_i$ : bandwidth of the link number  $i$ .  $Bp_i$  is the bandwidth of the path number  $i$ .
- (vi)  $w_{ik}$ : length of the link  $l_{ik}$ .
- (vii)  $TM_i$ : overall migration time on  $l_{ik}$ .

**3.2. Description of Our Replacement Method.** The concept splits the original migrated traffic into many flowlets, which are forwarded on several paths simultaneously to the new node in which the considered virtual service resource will be restored. All the flowlets have the same destination. Only the original virtual service resource is enabled to split migration traffic over the network, meaning that traffic is split once; no additional subdivision occurs until the flowlets reach the destination node. This question of separating migration traffic into several parts requires that we answer two questions: (1) how do we choose the best path for migration? and (2) which data migration mechanism should we use?

**3.2.1. Selection Criteria for the Best Migration Path.** Because our objective is to ensure a fast virtual service resource replacement, flows will be sent over the shortest paths from the source to the destination node. To overcome the limits of the Liu et al. [15] method which are related to the shortest migration path selection, our path selection method considers the bandwidth, link length, and the number of hops between the source and destination. Before selecting paths for the flowlets, we estimate the total migration time for each flowlet. We thus consider that (1) gives the migration time over a unit of distance, which means that, for link  $l_{ik}$ , the migration time is given by

$$d_{ik} = w_{ik} \cdot d_{\text{mig}}. \quad (2)$$

Thus, for each path  $P_i$  of the graph, the total migration time  $TM_i$  of the path number  $i$  is given by

$$TM_i = \sum_{k=1}^{n-1} d_{k(k+1)}. \quad (3)$$

The best migration path is that with the minimal migration time:  $\text{Min}(TM_i)$  ( $1 \leq i \leq n$ ), where  $n$  is the number of paths between the source and destination. While our replacement method uses multiple paths for migration, these paths are the best that comply with (3). Figure 6 illustrates this situation for the case of an incorrect virtual machine migration path selection encountered with [15] when bandwidths are equals. Considering that the memory transfer time is  $m_{\text{mig}} = 15$  min and the interruption time is  $i_{\text{mig}} = 1$  min, we obtain the following values for the replacement time:

- (i) Path 1:  $bp = 9$  Gbps and  $d_{\text{mig}} = 15/9 + 1 = 1.7 + 1 = 2.7$  min.  
 $d_{n1n2} = 1 \times 2.7 = 2.7$  min;  
 $d_{n2n5} = 2 \times 2.7 = 5.4$  min;  
 $d_{n5n7} = 2 \times 2.7 = 5.4$  min;  
 $d_{n7n8} = 1 \times 2.7 = 2.7$  min; the total migration time for

path 1 is  $TM_1 = 2 \times 2.7 + 2 \times 5.4 = 5.4 + 10.8 = 16.2$  min

- (ii) Path 2:  $bp = 8$  Gbps and  $d_{\text{mig}} = 15/8 + 1 = 1.9 + 1 = 2.9$  min.

$TM_2 = 3 \times 2.9 + 2.5 \times 2.9 = 8.7 + 7.25 = 15.95$  min

- (iii) Path 3:  $bp = 9$  Gbps and  $d_{\text{mig}} = 15/9 + 1 = 1.7 + 1 = 2.7$  min.

$TM_3 = 1 \times 2.7 + 2 \times 2.7 + 1 \times 2.7 = 10.8$  min

We note that  $TM_3 < TM_2 < TM_1$ . Therefore, in the case of a two-part flow, the best replacement paths are paths 3 and 2 because they provide the lowest replacement times. In the case of a single path migration approach, path 3 would be selected as the single best migration path, which is better than that of Liu et al. [15], which would also consider path 1 to be the best path due to its equal bandwidth.

Figure 7 shows how our migration path selection method overcomes the second limit of [15], which is related to the incorrect migration path selection when the path bandwidths are different.

When looking for the potential virtual service resource replacement time through each path, the following values are obtained:

- (i) Path 1:  $bp = 5$  Gbps and  $d_{\text{mig}} = 15/5 + 1 = 3 + 1 = 4$  min. The total migration time for path 1 is  $TM_1 = 2 \times 4 + 1 \times 4 + 1 \times 4 + 1 \times 4 + 2.5 \times 4 = 30$  min.

- (ii) Path 2:  $bp = 2$  Gbps and  $d_{\text{mig}} = 15/2 + 1 = 7.5 + 1 = 8.5$  min.  $TM_2 = 8.5 \times 2.5 + 8.5 \times 3 + 8.5 \times 2.5 = 68$  min.

- (iii) Path 3:  $bp = 3$  Gbps and  $d_{\text{mig}} = 15/3 + 1 = 5 + 1 = 6$  min.  $TM_3 = 1 \times 6 + 2 \times 6 + 1 \times 6 + 1 \times 6 + 1 \times 6 = 36$  min.

- (iv) Path 4:  $bp = 4$  Gbps and  $d_{\text{mig}} = 15/4 + 1 = 3.75 + 1 = 4.75$  min.  $TM_4 = 2 \times 4.75 + 1 \times 4.75 + 1 \times 4.75 + 2 \times 4.75 = 28.5$  min.

These values suggest that the potential best replacement paths are paths 1 and 4, which are better than the single path 1 selected by Liu et al. [15].

**3.2.2. Our Data Migration Mechanism.** The virtual service resource data migration, which is similar to virtual machine migration from one physical host to another, must maintain a consistent state of the service offered in the destination machine [23]. To do this, we must provide an appropriate data forwarding mechanism. Two primary mechanisms have been proposed in the context of virtual machine migration to date:

- (i) *Cold migration* (or “stop and copy”) [24] consists of stopping the virtual machine and copying its data to the destination node. Once the service is restored in the destination node, the virtual machine is run. The drawbacks of this method include a long downtime and bootstrapping on the destination node.
- (ii) *Live migration* [24, 25] includes three approaches: precopy, postcopy, and hybrid postcopy. The *precopy* approach mainly consists of transmitting all virtual

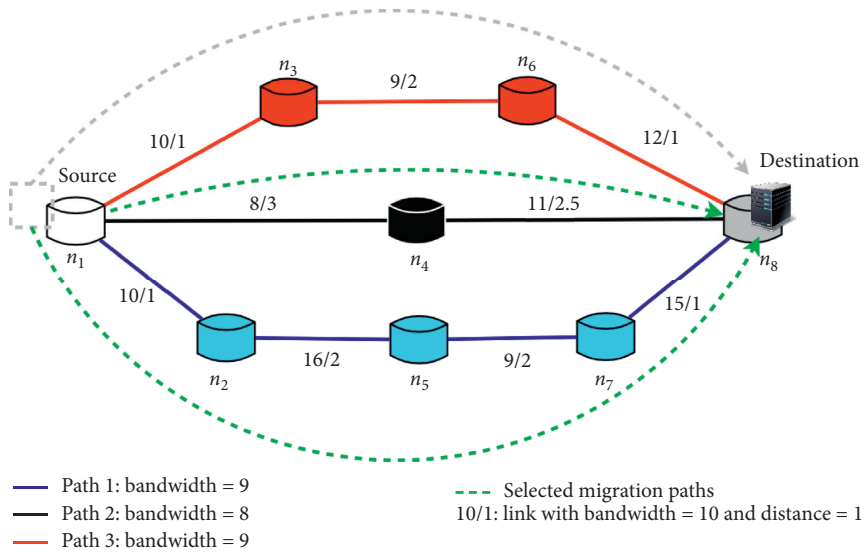


FIGURE 6: Best replacement paths selection for paths with equal bandwidths.

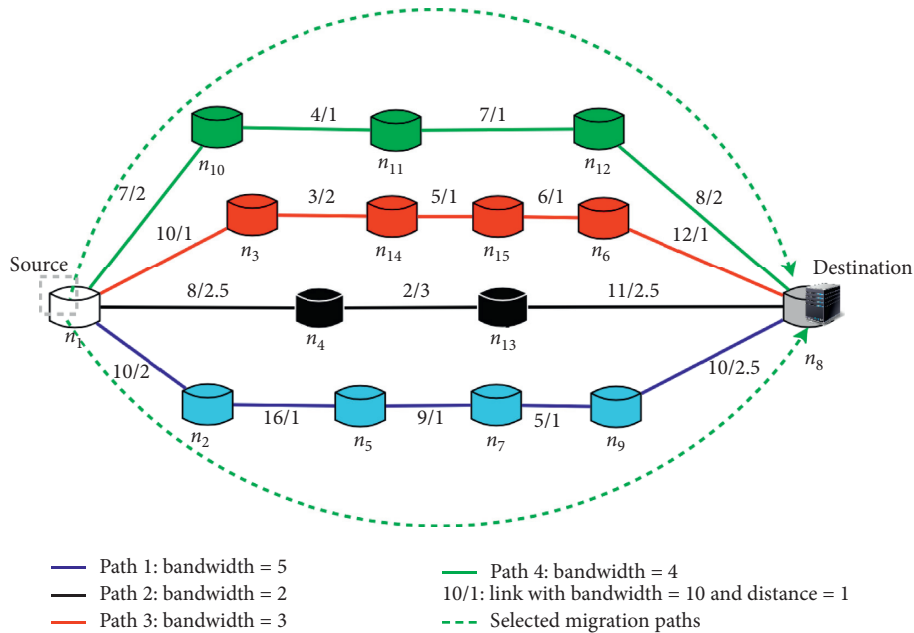


FIGURE 7: Best replacement path selection for paths with different bandwidths.

machine memory pages to the destination host while the service is running. After a given point, the service is stopped on the source host, and the rest of the modified memory pages are copied to the destination host. The main problem with this method occurs when the maximum interruption time is too low to send the last modified memory pages to the destination node [17]. In this case, the migration time can be high. Compared to the precopy migration approach, in *postcopy*, the virtual machine is stopped in the source node first. Then, data are copied, and the virtual machine is restored in the destination node. However, while the virtual

machine is directly started on the destination host in an inconsistent state, the failure of the source or destination node during migration causes the inevitable loss of state integrity in virtual machine memory. *Hybrid postcopy* [26] has been proposed to reduce postcopy performance issues. Progressively modified memory pages are transferred from the source node to the destination (precopy) without disrupting service (see Figure 8). The copy of the modified memory pages ends when a critical point is reached; the service is suspended on the source machine and its state is restored on the destination machine with consistent data (postcopy). We

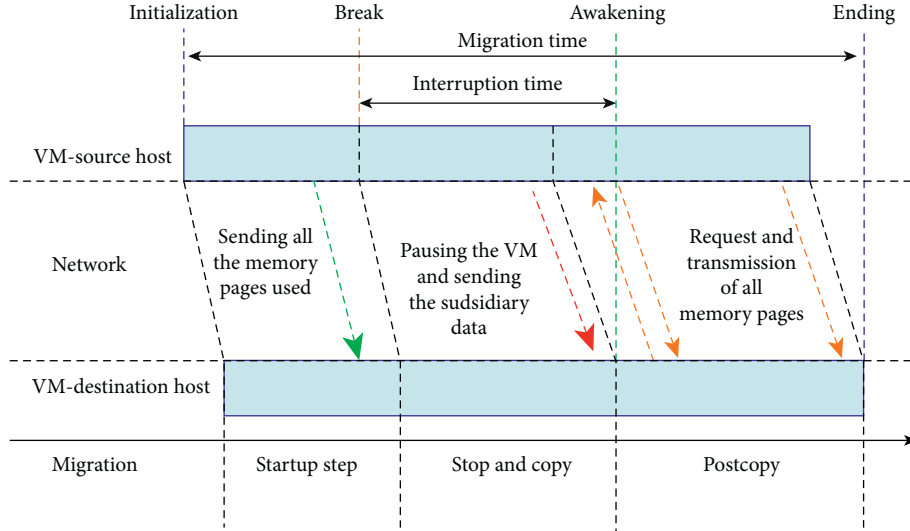


FIGURE 8: Hybrid postcopy algorithm.

propose this hybrid migration approach to substantially reduce both downtime and total migration time.

We adapt this hybrid virtual machine migration approach to our virtual service resource replacement method. In the precopy step, we transfer the gradually modified memory pages, which contains service data and traffic and requests the execution state, along the shortest paths between the source and destination.

#### 4. Our Enhanced Virtual Service Resource Replacement Method Using Traffic Splitting with Link Throughput as a Best Path Selection Criterion

In Section 3, we provide a flow-splitting-based virtual service resource replacement approach that uses bandwidth and link length as the criteria for shortest replacement path selection. In practice, however, data flows on links are not constant, as is bandwidth; thus, some paths can be selected as the shortest based on the bandwidth, while the flow in the corresponding links is not suitable for service data migration. This section presents our throughput and link length-based solution as the shortest path selection criteria for a virtual service resource replacement.

**4.1. Challenges of a Data Flow-Based Path Selection Decision.** The main difficulty related to data flow as a service data forwarding path criterion is its dynamicity, which is specific to each link and changes with network traffic. The following challenges must be addressed in this context:

- (i) How to identify the current link flow rate at each moment in time based on the traffic: to achieve this, a global view of the network and its traffic is required. Virtual service resources do not have this ability,

which is why we use SDN to surmount this challenge.

- (ii) How to make a reliable replacement path decision that ensures that flows are delivered within an acceptable time: a better path at a time  $t$  may not be a better path at time  $t + 1$  due to variations in flow data rate. To solve this problem, we use the throughput mean on each link to consider large variations in throughput.

**4.2. Our Data Flow-Based Path Selection Solution.** Considering the issue of network traffic mapping, the SDN controller uses its global network overview (i.e., topology, failures, and traffic) as a solution. The network accountability activity in the controller can provide this information. Network accountability is an activity that allows an infrastructure provider to check activity within a network to detect suspicious behavior or extract useful information for network profitability [27]. Thus, this accounting makes it possible to incorporate security and credibility [28] in the network. In our solution, when a virtual service resource replacement is required, the source node sends a specific message to the controller, mapping network traffic. This map will be used later to determine the best replacement paths.

Based on the network traffic map, the controller estimates the minimum ( $v_{\min}$ ) and maximum ( $v_{\max}$ ) data flow variations using observations made during a time period  $dt$ . Then, the controller computes the shortest paths using  $v_{\min}$  and  $v_{\max}$ . Thus, for a given link  $l_{ik}$ , the minimum and the maximum replacement times are, respectively, given by

$$d_{ik(\min)} = \frac{m_{\text{mig}}}{v_{\min}} + i_{(\text{mig})ik}, \quad (4)$$

$$d_{ik(\max)} = \frac{m_{\text{mig}}}{v_{\max}} + i_{(\text{mig})ik}, \quad (5)$$

where  $i_{(\text{mig})ik}$  is the latency time in the intermediate nodes.



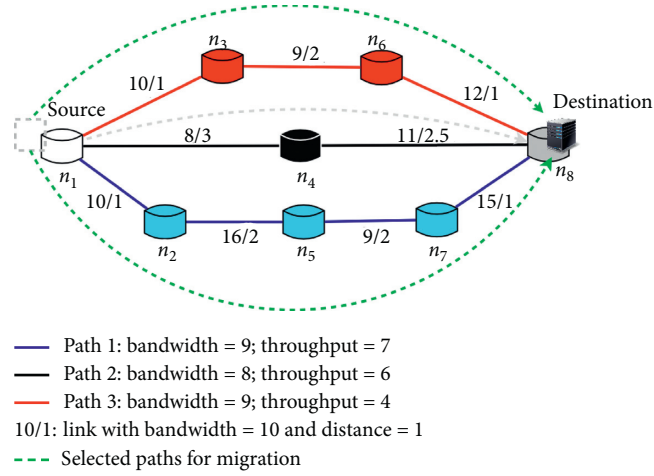


FIGURE 9: Best replacement path selection with throughput as one of selection criteria instead of bandwidth.

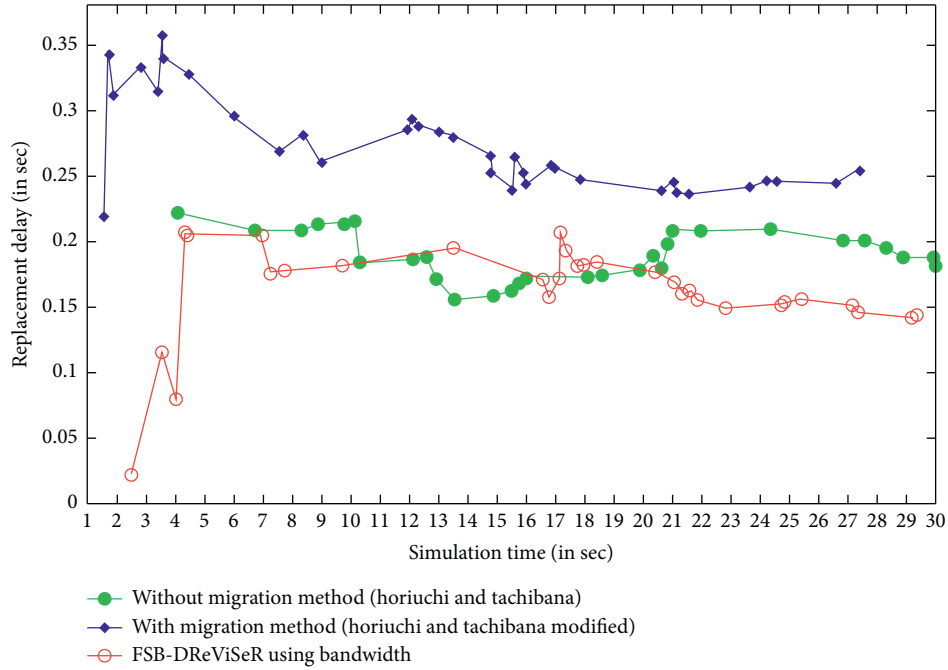


FIGURE 10: Replacement delays for a 20-node network.

From (3)–(5), we conclude that, for a path  $P_i$ , the minimum and maximum total replacement times are, respectively, given by

$$TM_{i(\min)} = \sum_{k=1}^{n-1} d_{k(k+1)\min}, \quad (6)$$

$$TM_{i(\max)} = \sum_{k=1}^{n-1} d_{k(k+1)\max}, \quad (7)$$

where  $d_{k(k+1)\min} = (m_{\text{mig}} / \prod_{j=1}^n \text{Min}_{v_{(\min)j}}) + i_{(\text{mig})ik} \cdot \prod_{j=1}^n \text{Min}_{v_{(\min)j}}$  of a given path  $P_i$  is the minimum flow data rate of this path, which is given by the throughput of the link with the minimum bandwidth.

A replacement path selection approach based on  $TM_{i(\min)}$  would not handle cases where the flow data rate is higher. Similarly, a selection based on the maximum replacement time  $TM_{i(\max)}$  would fail to deal with cases where the flow data rate is higher. We suggest using the flow average given by

$$d_{ik} = \frac{d_{ik(\min)} + d_{ik(\max)}}{2}. \quad (8)$$

This average should reduce instances of nonconsidered link flow data rate.

Figure 9 shows how the throughput can be used as a migration path selection criterion instead of bandwidth. In this

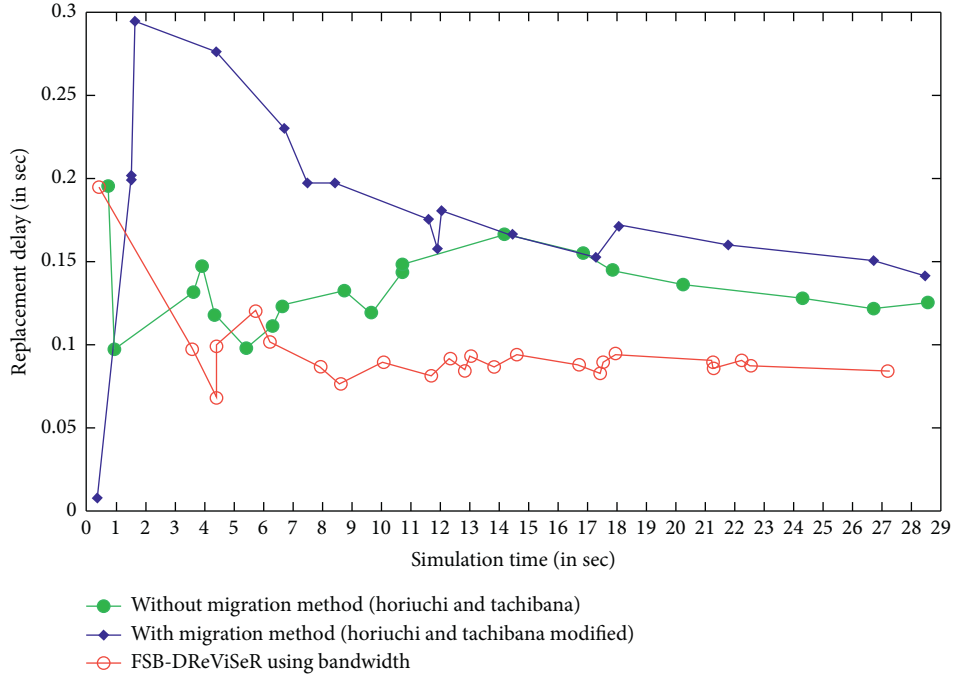


FIGURE 11: Replacement delays for a 60-node network.

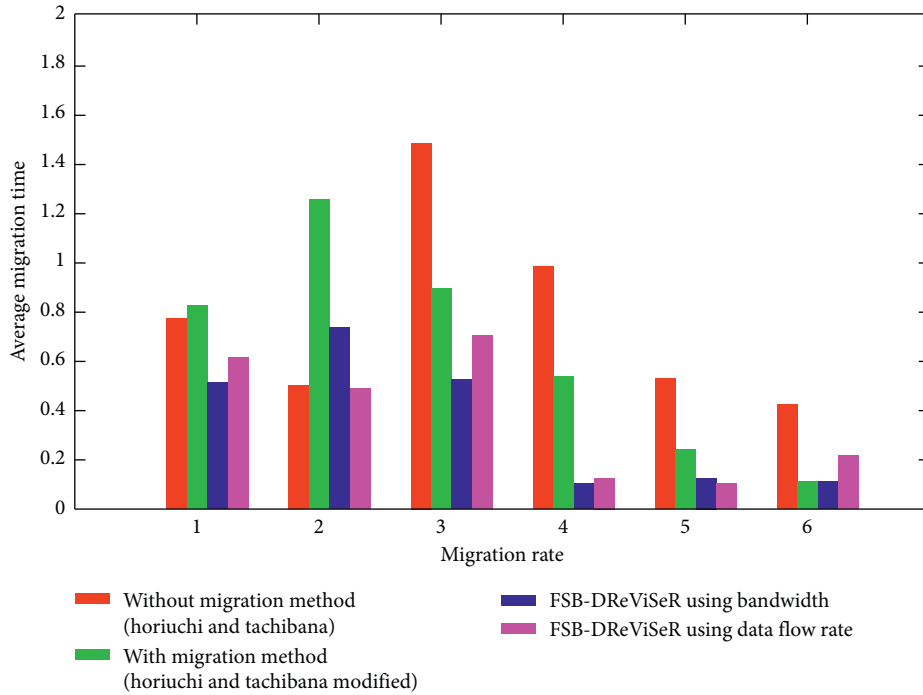


FIGURE 12: Average migration time versus migration rate for a 20-node network.

figure, the average throughput of each path is considered, and the different replacement times are computed as follows:

(i) Path 1:  $bp = 9$  Gbps and  $d_{mig} = 15/7 + 1 = 2.14 + 1 = 3.14$  min.  $d_{n1n2} = 1 \times 3.14 = 3.14$  min;  $d_{n2n5} = 2 \times 3.14 = 6.28$  min;  $d_{n5n7} = 2 \times 3.14$

$= 6.28$  min;  $d_{n7n8} = 1 \times 3.14 = 3.14$  min; the total migration time for Path 1 is  $TM_1 = 2 \times 6.28 + 2 \times 3.14 = 18.84$  min.

(ii) Path 2:  $bp = 8$  Gbps and  $d_{mig} = 15/6 + 1 = 2.5 + 1 = 3.5$  min.  $TM_2 = 3 \times 3.5 + 2.5 \times 3.5 = 10.5 + 8.75 = 19.25$  min.

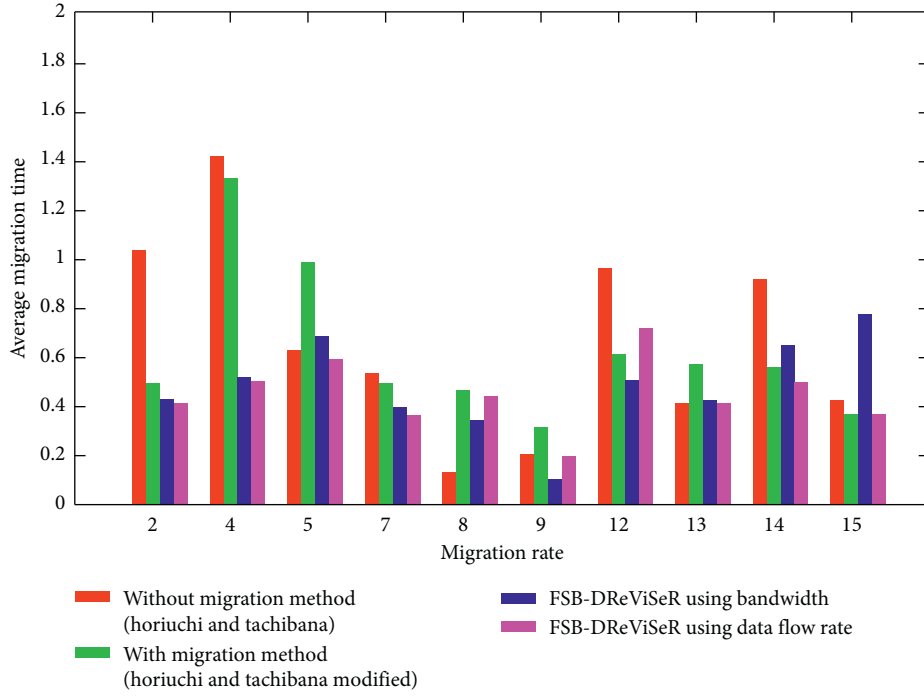


FIGURE 13: Average migration time versus migration rate for a 60-node network.

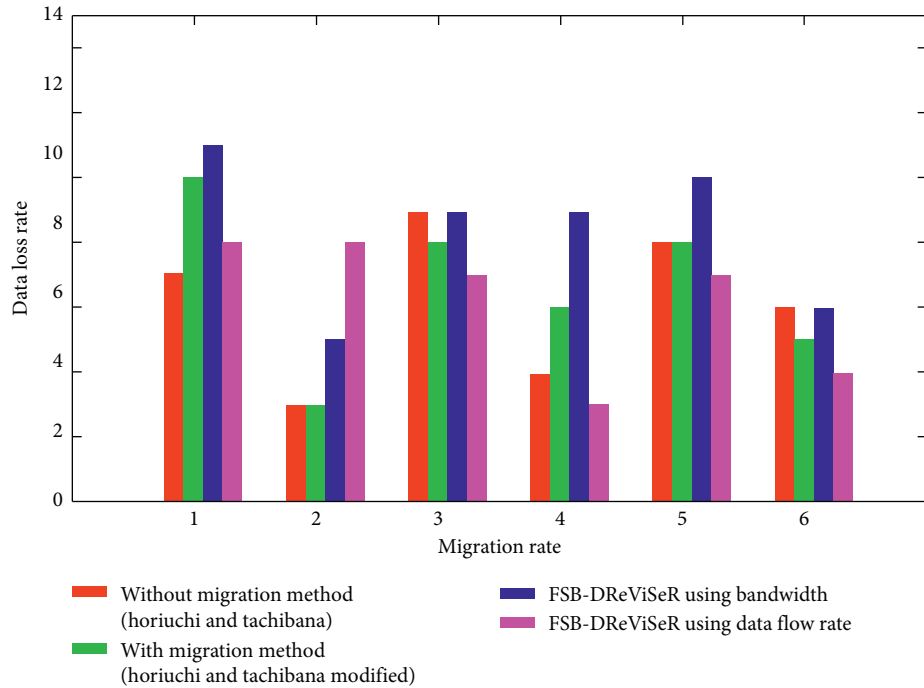


FIGURE 14: Data loss rate versus migration rate for a 20-node network.

(iii) Path 3:  $bp = 9 \text{ Gbps}$  and  $d_{\text{mig}} = 15/4 + 1 = 3.75 + 1 = 4.75 \text{ min.}$   
 $TM_3 = 1 \times 4.75 + 2 \times 4.75 + 1 \times 4.75 = 12.56 \text{ min.}$

With a two-parted flow, the best replacement paths are paths 3 and 1, not paths 3 and 2 as is determined when bandwidth is used rather than the throughput.

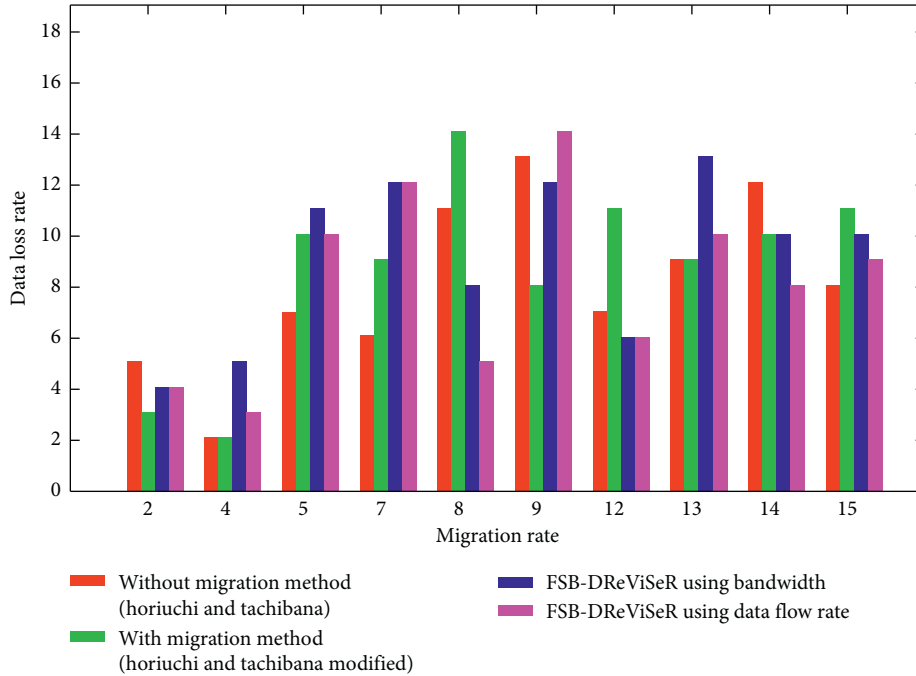


FIGURE 15: Data loss rate versus migration rate for a 60-node network.

## 5. Simulations Results and Discussion

To evaluate the performances of our approaches, we ran simulations, the results of which are presented in this section. In Section 5.1, we discuss the influence of flow splitting on virtual server resource replacement using bandwidth as a criterion for replacement path computation. Section 5.2 presents a comparison of the migration path selection approach using bandwidth and that using flow data rate. Simulations were performed in the OMNET++ discrete event network simulator, version 5.0. These simulations were performed on a small and large network to describe the performance of our approaches on different sized networks: the small network is network 1 (20 nodes and 31 links), and the large network is network 2 (60 nodes and 90 links).

*5.1. Impact of Flow Splitting Using Bandwidth as a Path Choice Criterion on QoS.* In this section, we compare the replacement delays of a virtual service resource using our FSB-DReViSeR method (FSB-DReViSeR using bandwidth) based on graphs, the approach of Horiuchi and Tachibana [8] without a migration technique (replacement delay without a migration method (Horiuchi and Tachibana)), and that with a migration technique (replacement delay with a migration method (Horiuchi and Tachibana modified)). Figures 10 and 11, respectively, present the results obtained for network 1 and network 2. The migration technique used for the approach in [8] is the hybrid virtual machine migration method [26]. Simulations were performed for each method, and the average replacement delays for each approach were collected.

In the small network, the replacement delays of our replacement method (FSB-DReViSeR using bandwidth) are shorter than those of Horiuchi and Tachibana. This can be explained by the fact that our approach uses the best pre-calculated paths and splits the replacement traffic. However, in some rare cases, such as between the simulation times  $t = 11$  s and  $t = 16$  s in network2, the approach in [8] yields lower replacement delays than our method. This result can be explained by the fact that the replacement traffic handled by the single path approach of Horiuchi and Tachibana did not require separation.

In addition, our bandwidth FSB-DReViSeR replacement approach offers more interesting replacement times than the two versions of the approach in [8] based on a tree topology. Another element that increases the replacement delay with the Horiuchi and Tachibana approach is the construction time of a minimum spanning tree from a graph topology before replacement. The use of a single path in the approach in [8] to perform the service migration also contributes to increasing this overall replacement delay.

*5.2. Impact of Flow Splitting Using Data Flow Rate as a Path Selection Criterion on QoS.* We evaluate this impact by comparing the performances of our FSB-DReViSeR approach using link throughput (FSB-DReViSeR using flow data rate) as a criterion for selecting replacement paths with that using bandwidth (FSB-DReViSeR using bandwidth). To better approximate reality, we randomly induced variations in the data flow based on a uniform law (0, 1). The average migration times were collected based on the number of replacements performed to determine which approach offers the best QoS, even with high traffic.

In a small network (see Figure 12), the bandwidth-focused FSB-DReViSeR yields shorter replacement delays in general. However, the difference from the throughput-focused FSB-DReViSeR is not significant when dealing with a large network (see Figure 13). This result could be explained by the fact that traffic is more important in large networks, which causes the link throughput to generally approach the link bandwidth. Thus, even after calculating variations between  $v_{\min}$  and  $v_{\max}$ , we obtain values close to the bandwidth in most cases. This result suggests that bandwidth-focused FSB-DReViSeR and throughput-focused FSB-DReViSeR have similar performances when the size of the network is large (i.e., more than 60 nodes).

**5.3. Impact of Flow Splitting on Service Consistency after Replacement.** A problem with flow splitting strategies is reassembling the flow at the destination, as mentioned in Section 2. Thus, the splitting rate must be reduced as much as possible. In this section, we evaluate the performance of our approach in relation to its influence on the consistency of the service after virtual service resource replacement. To address this, we are interested in the data migration lost rate, which we compare with the number of replacements. We used the two-part flow splitting version to minimize the number of splitting incidences.

Figures 14 and 15 provide an overview of the results obtained. We notice that, in small (see Figure 14) and large networks (see Figure 15), the data loss rate is generally lower with the two versions of the Horiuchi and Tachibana method than ours. The flow splitting method that we use with our migration technique is responsible for this result. However, in a large network, our approach yields a lower packet loss rate when the number of resource replacements is important. Our throughput-focused FSB-DReViSeR method also generally provides better replacement delays than the bandwidth-focused FSB-DReViSeR method, even if equivalent points exist.

The results presented show that our flow splitting-based dynamic virtual service resource replacement approach surpasses the approach in [8] in several ways (e.g., replacement delay and data loss rate), especially when the network is large, and the number of replacements is important. Furthermore, among the two proposed approaches, the method that considers the throughput as a criterion for selecting the best migration path (the throughput-focused FSB-DReViSeR) yields better performance.

## 6. Conclusion

Our objective in this paper was to improve the QoS in virtual heterogeneous networks by managing traffic variations due to user mobility. We thus proposed two approaches for dynamic virtual service resource replacement based on flow splitting: the first, called bandwidth-focused FSB-DReViSeR, uses bandwidth and network link length as key criteria to select the fastest replacement paths; the second, called throughput-focused FSB-DReViSeR, uses link throughput and length as key criteria. Simulations performed on small

and large networks showed that the throughput-focused approach is more credible and reliable. Comparisons with other approaches, such as that of Horiuchi and Tachibana, also showed that our replacement method better satisfies QoS requirements in a graph network, which is closer to reality than that of the Horiuchi and Tachibana approach, which is based on a tree topology. Future work should address the unavailability of a backup server for the virtual service resource replacement in a given virtual plane.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] N. C. Fernandes, M. D. Moreira, I. M. Moraes et al., "Virtual networks: isolation, performance, and trends," *Annales des Télécommunications*, vol. 66, no. 5-6, pp. 339–355, 2011.
- [2] I. Lee and K. Lee, "The internet of things (IoT): applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.
- [3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [4] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013.
- [5] M. M. A. Khan, N. Shahriar, R. Ahmed, and R. Boutaba, "Multi-path link embedding for survivability in virtual networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 253–266, 2016.
- [6] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding in optical datacenter networks," *Journal of Optical Communications and Networking*, vol. 7, no. 12, pp. 1160–1171, 2015.
- [7] N. Shahriar, R. Ahmed, A. Khan, S. R. Chowdhury, R. Boutaba, and J. Mitra, "ReNoVatE: recovery from node failure in virtual network embedding," in *Proceedings of the 2016 12th International Conference on Network and Service Management (CNSM)*, pp. 19–27, IEEE, Montreal, Canada, October 2016.
- [8] S. Horiuchi and T. Tachibana, "Dynamic replacement of virtual service resources based on tree topology for mobile users in virtual networks," *Journal of Computers*, vol. 13, no. 12, pp. 1335–1348, 2018.
- [9] Y. Koyanagi and T. Tachibana, "Dynamic resource allocation based on amount of traffic in virtual networks," Technical report, IEICE, Miyazaki, Japan, 2014.
- [10] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: methods and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, 2015.
- [11] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *IEEE Internet Computing*, vol. 17, no. 2, pp. 20–27, 2012.
- [12] P. Lechowicz, S. Spadaro, R. Rumipamba-Zambrano, K. Walkowiak, and J. Perelló, "Migration planning from

- elastic optical networks to spectrally-spatially flexible optical networks,” in *Proceedings of the 2018 Photonics in Switching and Computing (PSC)*, pp. 1–3, IEEE, Limassol, Cyprus, September 2018.
- [13] X. Li, L. Tan, and F. Li, “Optimal cloud resource allocation with cost performance tradeoff based on internet of things,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6876–6886, 2019.
- [14] J. F. Myoupo, Y. F. Yankam, and V. K. Tchendji, “On the dynamic replacement of virtual service resources for mobile users in virtual networks,” *Journal of Computers*, vol. 15, no. 1, pp. 10–21, 2020.
- [15] J. Liu, Y. Li, and D. Jin, “SDN-based live VM migration across datacenters,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 583–584, 2015.
- [16] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: from concept to implementation,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.
- [17] V. Kherbache, “Ordonnancement des migrations à chaud de machines virtuelles,” Doctoral dissertation, Université Côte d’Azur, Nice, France, 2016.
- [18] V. K. Tchendji, Y. F. Yankam, and J. F. Myoupo, “Conflict-free rerouting scheme through flow splitting for virtual networks using switches,” *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 13, 2018.
- [19] J. Leino and J. Virtamo, “Insensitive traffic splitting in data networks,” in *Proceedings of the 19th International Teletraffic Congress (ITC19)*, pp. 1355–1364, Beijing, China, August 2005.
- [20] T. S. Pham, J. Lattmann, J. L. Lutton, L. Valeyre, J. Carlier, and D. Nace, “A restoration scheme for virtual networks using switches,” in *Proceedings of the 2012 4th International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 800–805, IEEE, St. Petersburg, Russia, October 2012.
- [21] M. S. Seddiki, “Allocation dynamique des ressources et gestion de la qualité de service dans la virtualisation des réseaux,” Doctoral dissertation, Université de Lorraine, Nancy, France, 2015.
- [22] J. Veerasamy, S. Venkatesan, and J. C. Shah, “Effect of traffic splitting on link and path restoration planning,” in *Proceedings of the 1994 IEEE GLOBECOM. Communications: The Global Bridge*, pp. 1867–1871, San Francisco, CA, USA, 1994.
- [23] F. Zhang, G. Liu, X. Fu, and R. Yahyapour, “A survey on virtual machine migration: challenges, techniques, and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1206–1243, 2018.
- [24] U.. Keshavamurthy and H. S. Guruprasad, “VM migration: a survey,” *Global Journal of Engineering Science and Researches*, vol. 2, pp. 219–224, 2015.
- [25] C. Clark, K. Fraser, S. Hand et al., “Live migration of virtual machines,” in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, vol. 2, pp. 273–286, Berkeley, CA, USA, 2005.
- [26] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, “Live and incremental whole-system migration of virtual machines using block-bitmap,” in *Proceedings of the 2008 IEEE International Conference on Cluster Computing*, Tsukuba, Japan, September 2008.
- [27] E. Keller, R. B. Lee, and J. Rexford, “Accountability in hosted virtual networks,” in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, Barcelona, Spain, August 2009.
- [28] L. Mekouar, Y. Iraqi, and R. Boutaba, “Incorporating trust in network virtualization,” in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology*, Bradford, UK, June 2010.