Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2022 SCPE. Volume 23, Issues 4, pp. 421–438, DOI 10.12694/scpe.v23i4.2010

A DRIVEN MODERN PORTFOLIO THEORY FOR VIRTUAL NETWORK EMBEDDING IN SDN-ENABLED CLOUD

ABDERRAHIM BOUCHAIR,[*] BELABBAS YAGOUBI [†] AND SID AHMED MAKHLOUF [‡]

Abstract. Network virtualization (NV) has evolved systematically through the urge to share computing resources and improve service deployment in a large-scale environment. Virtual network embedding (VNE) is a well-established technology applied to reinforce the NV process, providing a devoted implementation for a particular case study. In cloud computing, integration of software-defined networking (SDN) has proved to be a practical support to the principal cloud utilities. In return, the SDN-enabled cloud offers innovative deployment techniques for network-based services, which increase the opportunity to efficiently incorporate new network management policies that solve the VNE problem. In this paper, the authors proposed a transition of modern portfolio theory (MPT) into a VNE approach that optimally addresses the selection and ranking of resources in data center networks (DCNs). Results analysis demonstrates the VNE approach's better performance versus similar methods in terms of acceptance ratio, runtime, and substrate resource utilization.

Key words: Cloud computing, Virtual network embedding, Portfolio selection, SDN, Data center network, Allocation policy, Resource importance.

AMS subject classifications. 68T14, 18B15, 90C59, 91G10

1. Introduction.

Since its initial launching point, cloud computing kept seeking to proffer as closely as possible the ideal design for high-end resource sharing scalability. Extending on-demand services is the main task of every cloud provider [1]. This is relevant to resource availability placed in physical infrastructure known as the data center (DC). A DCN allows connection between many DC-based servers, guaranteed by offering traffic reliability through efficient network equipment and several virtualization support techniques [2]. The practice of cloud virtualization across DCNs has typically opened new research challenges related to global network performance. Furthermore, various virtualization strategies have been introduced by engaging different types of full or partial virtualization in a collaborative scheme. Thus, VNE derives from finding relationships between virtual networks (VNs) and substrate networks (SNs) that would lead to see the cloud system as a congregate physical structure. The challenge of resource assignment in NV is commonly known as the VNE problem [3]; it primarily consists of mapping virtual resources onto physical devices. Typically, given virtual resources to be embedded are settled as a communication channel of virtual nodes (VNods) interconnected via virtual links (VLins), forming a virtual network request (VNR).

The emergence of SDN back in 2008 has brought outstanding flexibility and robust manageability to traditional networks, resulting from the novel implementation of OpenFlow protocol. Therefore, SDN conception addresses the controllability issue in DCNs by separating different networking stack components into three fundamentally layers: application plane, control plane, and data plane. SDN communications have two divisions crossed at the controller: Firstly, the transfer of information between application and control planes; Afterward, the management servers in the control plane require a pre-set up logic to send instructions to the data plane [4]. The Data plane contains the underlying networking devices like switches and routers to forward the commands.

During every advanced case study, it is inevitable to test future work performances, similar to the prediction routine of stock market behavior in finance. In 1990, Harry Markowitz awarded the Nobel Memorial Prize in

[*]L.I.O. Laboratory, Department of Computer Science, University of Oran1 Ahmed Ben Bella, P.O. Box 1524 El M'Naouer, Oran, Algeria(bouchair.abderrahim@gmail.com).
[†]L.I.O. Laboratory, Department of Computer Science, University of Oran1 Ahmed Ben Bella, P.O. Box 1524 El M'Naouer, Oran, Algeria(byagoubi31@gmail.com).
[‡]L.I.O. Laboratory, Department of Computer Science, University of Oran1 Ahmed Ben Bella, P.O. Box 1524 El M'Naouer, Oran, Algeria(sidahmed.makhlouf@gmail.com).

Economic Science for developing the portfolio choice theory. Markowitz model or portfolio theory (PT) [5] is a mathematical model that considers a portfolio optimization based on two angles: a preferable mean return of a portfolio and an unwanted return denoted as a portfolio risk or variance.

In this work, we attempt to layout a novel node ranking approach for VNE. This challenge adopts MPT analysis as a mapping model for the VNE problem, by resolving these couple significant issues:

(i) What is the optimal selection order of a substrate node (SNod)/link (SLin) for a given VNod/VLin to embed ?
(ii) How to reduce resource utilization rate ?

MPT helps sharpen the supervision quality of the SDN controller and provides a fordable NV running cost by identifying the most suitable substrate resource. In this paper, we explore the most realizable solutions for prior mentioned problematics by introducing the following contributions:

- We propose an analytical solution to solve the VNE problem by projecting the MPT model to adapt to the VNE standards. The MPT model provides a decision support optimization, guaranteed in both SNs and VNRs.
- We suggest using the term *Composite*, a compound element of node and its attached links to facilitate the mapping process in a single stage.
- Employ the obtained composite to implement an effective matching mechanism, which will optimize the resource allocation. This optimization is devoted to overcoming the limitation of MPT liability in dealing with the composite provisioning.
- A set of simulation experiments are conducted to validate the accuracy of the proposed approach.

In the remaining of this paper, we present in Sect. 2, the most associated contributions to our work. Next, an outlined mathematical model of MPT, its projection on our approach, and problem description are presented in Sect. 3. In Sect. 4, we introduce system architecture and the implementation of the proposed VNE policy. Section 5 exhibits the experimental setup and performance evaluation. Finally, a highlighted conclusion is featured along with prospective studies.

2. Related Work. VNE problem remains undoubtedly an NP-hard problem due to the nodes and links constraints [3]. Many studies have drawn much to this matter, specifically those focusing on the issues crossed in cloud computing. The authors in [6] proposed a heuristic VNE algorithm based on defining the topology potential (VNE-TP), by developing a multi-objective optimization approach as an integer linear programming (ILP) model to solve the VNE problem. This novel approach relies on two coordinated stages, regarding node importance calculation and bandwidth function for VLins. The performance evaluations proved an improvement in VNR acceptance ratio and revenue ratio compared with two other similar heuristics. Nevertheless, increments in revenue ratio would generate a lack of VNE accuracy due to the two stages process. In this paper, we consider an ideal VNE process of one coordinated stage mapping, with an enhanced matching mechanism that deals with node and link mapping as one component to be processed.

The work depicted in [7], a metaheuristic-based approach is proposed to solve the VNE problem using a constructive particle swarm optimizer (CPSO-VNE), combined with a step-by-step heuristic scheme dependency. This latter collects the necessary path information from SN, adding more deep processing tasks (matrix calculations with probabilities and discrete vectors). Eventually, additional resource utilization is required to minimize time response, especially for online VNRs. However, in our study, the SDN controller replaces the work of the heuristic solution by offering agile network automation for the MPT measurements, and an ongoing reduced cost while centralizing the necessary operations.

Based on an ILP model, an exact VNE algorithm is proposed in [8] to increase the VNRs acceptance rate through location constraints between the VNRs requirements and the available resources in SN. The result analysis illustrates a better acceptance ratio with 15% more than previous heuristics. Notably, this approach does not support path-splitting, which successfully reduced the harnessing of substrate resources. Alternatively, not adopting a robust network architecture and simply generating random networks using the waxman model [9] does not correspond to the currently updated network technologies. Accordingly, we implemented the most widely-used DCNs to sustain a groundwork reference to a real case scenario in cloud practices.

In [10], the authors got inspired by a biological principle called Yuragi to propose a novel VNE method. Yuragi is a Japanese word which indicates a small disturbance generated extremely and internally. This work's main idea is to implement a mechanism that adapts organisms formulated as an attractor selection model for software-defined infrastructure (SDI). Performance evaluations illustrate the adaptability of finding a VNE solution under end-to-end delay behaviors, considering several node attributes. The authors have found a lack of access administration and management. Similarly to MPT-VNE, Yuragi-VNE demonstrates a positive application of adopting a solution to the VNE problem from other science fields.

SDN-based techniques promote various solutions to overcome the issues encountered in the rapid scaling infrastructures [11]. In [12], authors proposed a VNE algorithm targeting node selection with the most feasible embedding state, based on clustering coefficient information and node importance for ranking. The authors implemented a non-coordinated two-stage mapping algorithm, which accrues an extensive amount of calculations using the breadth-first search (BFS) algorithm for the node mapping stage and K-shortest algorithm for the link mapping stage. Moreover, the VNE algorithm's final part is based on collecting the outputs of the node-sorting algorithm and the algorithms for each of the two-stages, which occurs a higher time complexity than the Markowitz mean-standard deviation model.

The work accomplished in [13] exemplifies a proper implementation of an abstraction strategy to embed a clustered VNs in edge servers. This approach uses the SDN controller that applies a modified heapsort algorithm to find the pod with the most available resources. Therefore, authors have formulated the resource allocation process as an ILP problem. Experimental results revealed that the proposed work outperformed similar existing algorithms.

Unlike previous research works, our work seeks to maintain a rational background to the real case scenarios in cloud practices. In this work, we try to bypass the high-cost node ranking methods, including massive task implementations and any potential re-embedding process. This is done by employing a simplified node ranking metric inspired by the trading strategy in finance, which will maximize the substrate resource utilization for every mapping request. Thereby, we propose an adapted MPT as a VNE technique related to the investing concept of Markowitz.

3. Modern portfolio theory and problem formulation.

This section presents our work's principal sources, including an overview of MPT functionality and problem formulation.

3.1. Modern portfolio theory.

The main thrust of MPT is promoting the risk tolerance to build an optimal portfolio of assets with a maximum expected return and low-risk rate. Markowitz first introduces this risk-return relationship published an essay in 1952, and like any portfolio analysis, MPT typically consists of these two steps:

1. **Define the opportunity framework:** A set of particular assets or securities (Portfolio) available to the investor and their properties (mean and standard deviation).
2. **Portfolio selection:** Choose the optimal portfolio that is the most suited to investor requirements.

According to Markowitz, portfolio theory has the following pre-fixed assumptions to make it more applicable:

- Investors' decision (portfolio selection) is rational and based on the mean-variance concepts over a holding period.
- Risk-averse must be the frequent target of all investors.
- Investors focus on maximizing the profit rate for a given level of risk.
- Ability to determine an efficient set of assets or securities in a portfolio to get the most feasible solution (higher return for a given risk).
- All investors are aware beforehand with the sharing probability of return investment.

Based on our perspective setup, we identify the necessary points to achieve a valid projection from the Markowitz model to the VNE problem. Each asset from a given portfolio is characterized by an expected return and expected variance or standard deviation. The portfolio expected return (Ep) in Eq. 3.1 is calculated by multiplying the asset return r with its weight w.

$$Ep = \sum_{i=1}^{N} (w_i \times r_i) \tag{3.1}$$

$$\sum w_i = 1 \tag{3.2}$$

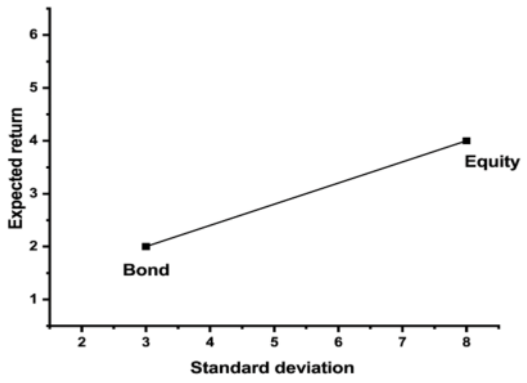

Fig. 3.1: Risk-return trade-off of a portfolio assets

The weight w_i represent the amount of money the investor is willing to spend on the i-th asset. The budget constraint in Eq. 3.2 is the sum of the weights, which must imperatively be equal to 1 or 100% if it is in percentage values (e.g., an investor can divide his capital equally on a portfolio of two assets $a1$ and $a2$ with 50% each).

Throughout the MPT optimization process, the portfolio standard deviation depends on calculating each asset's standard deviation. A high standard deviation portfolio indicates more volatility with the asset returns, and hence it is determined as a highly risky portfolio. Contrariwise, a portfolio with less volatility and more constancy is linked to a low standard deviation rate. The covariance measurement is also included as a diversification tool that can be used to add the assets that reveal negative covariance to reduce the overall volatility of a portfolio and ensure the risk-averse assumption. A positive covariance implies that two securities evolve in tandem; on another side, the covariance is negative when they evolve inversely. The portfolio assets can be displayed across a two-dimensional plane.

The risk-return relationship follows the bond asset's traced line representing low-risk low-return until the Equity asset, which indicates high-risk high-return. From Fig. 3.1, we can deduce that if an investor seeks a high return, he will be more likely to obtain an asset with a higher risk. MPT evaluates the investment impact on the overall portfolio risk and reward. However, our proposed method mainly focused on the assets management.

3.2. Analogy Context. VNE deals with the assignment of interconnected VNods that forms a VNR. Multiple VNods from a single VNR cannot be allocated to the same SNod, while a VLin can be mapped onto several SLins by creating a logical path through SNod, which are considered as intermediaries between the source and the destination. Besides, a single SNod can host several VNods from different VNRs. Based on this VNE standardization, we have found driven indigency to adopt a mutual representation for SNs and VNs elements denoted as Composites. A Composite exemplifies a collected node and its attached links in one element.

From Fig. 3.2, we can observe that the VLin between VNods E and F is passed through a SC that is considered as a hidden hop. Hence, VLin was embedded in two SLins. Composites have the role of simplifying the employment of the coordinated one-stage mapping in our VNE method. The primary concern is to set up MPT as a VNE-oriented strategy. From a VNE perspective, a DCN with its available resources represents a portfolio of N assets or shares (i.e., every substrate or virtual composite (SC or VC) has an asset as an equivalent). In order to apply the Markowitz mean-standard deviation model, we define the importance-standard deviation model as a projection of MPT optimization into a VNE problem, as shown in Table 3.1.

3.2.1. Composite importance. Fundamentally, a node's importance is determined according to its role to satisfy the global objective in a network-based approach. In our VNE method, we define the composite importance (Imp_c) in Eq. 3.3 based on the composite capacity (C_{cap}). This latter corresponds to the return

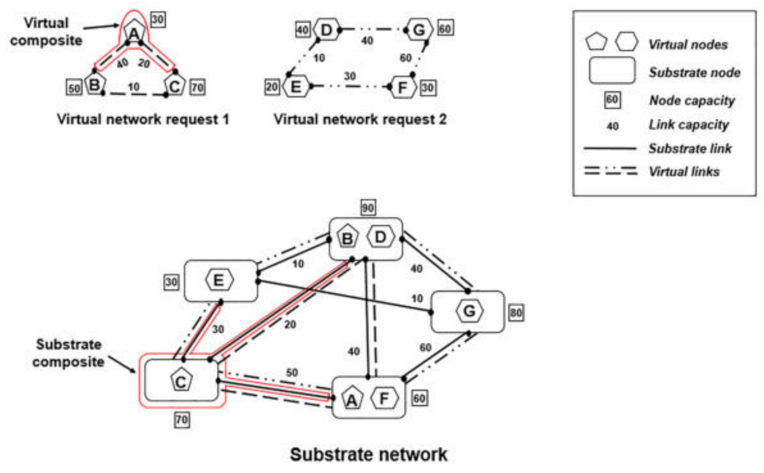

Fig. 3.2: Composites

Table 3.1: Similarities between MPT and VNE

MPT	VNE
Asset	Composite
weight	latency or bandwidth
Portfolio	DCN
Asset capital (money)	Composite capacity
Expected portfolio	Importance
Investing	Embedding or allocation

measurement of a single asset adapted from Eq. 3.1.

$$Imp_c = w \times C_{cap} \tag{3.3}$$

where w is the composite weight depends on the dedicated mean latency or bandwidth for each attached link. In the case of a SC, the weight is related to the latency property of a physical server's attached links.

$$SC_w = \frac{1}{\sum la_i} \tag{3.4}$$

where $\sum la_i$ is the total latency of the attached subtract links in the i-th SC. The reason behind choosing latency as a composite weight returns to its critical impact on the DCN connectivity. Thereby, a lower latency evinces the priority of a given composite. We note VC_w as VC weight related to the attached links' bandwidth of a virtual machine (VM).

$$VC_w = \frac{\sum Bw_j}{\sum Bw_{VN}} \tag{3.5}$$

where $\sum Bw_{VN}$ the total bandwidth of a VN, and $\sum Bw_j$ denotes the sum of the allocated bandwidth for the attached links in the j-th VC. In our approach, the latency is predefined during the creation of a DCN, yet it cannot be specified in case of a VNR until the VN is running, which left us only to consider VLin bandwidth property as its exact weight.

The return factor in MPT is settled on the amount of money granting for each asset, while the expected return is the total money the investor will probably earn after purchasing some assets from a stock market. However, from a resource allocation perspective, VNE deals with the VNR computational requirements on the one side and with SN's ability to assure the physical resources' availability and reliability on the other side. In light of this, we consider the relative SC capacity to the total network capacity as the main profit for both service providers (SPs) and cloud users.

$$RSC_{cap} = \frac{\sum SC_{cap}}{\sum SN_{cap}} \tag{3.6}$$

where $\sum SC_{cap}$ the total SN capacity. $\sum SN_{cap}$ denotes the sum capacity of the local resources of a physical server:

$$SC_{cap} = \sum(CPU, RAM, Storage, Bw) \tag{3.7}$$

In case of a VNR, the relative VC capacity is calculated in this way:

$$RVC_{cap} = \frac{\sum VC_{cap}}{\sum VN_{cap}} \tag{3.8}$$

where $\sum VN_{cap}$ the total of capacity of VN. $\sum VC_{cap}$ denotes the sum capacity of the local resources of a VM:

$$VC_{cap} = \sum(CPU, RAM, Size, Bw) \tag{3.9}$$

We note that the final value of any composite capacity is normalized using the following feature scaling in Eq. 3.10:

$$FC_{cap} = \frac{C_{cap} - MinC_{cap}}{MaxC_{cap} - MinC_{cap}} \tag{3.10}$$

where $MaxC_{cap}$ and $MinC_{cap}$ denote the maximum and minimum composite resource capacity, respectively, from a set of composites in network. Since C_{cap} features the return aspect from MPT; FC_{cap} values are between 0 and 1, which simplifies the data sorting.

3.2.2. Composite standard deviation. Principally, the standard deviation of a composite σ_c is the square root of the composite variance σ^2.

$$\sigma_c = \sqrt{\sigma_C^2} \tag{3.11}$$

$$\sigma_C^2 = \frac{(FC_{cap} - Imp_C)^2}{K} \tag{3.12}$$

We note that K is related to the VNR size V and SN size S. The variance of VC is calculated in the following way:

$$\sigma_{VC}^2 = \frac{(FC_{cap} - Imp_{VC})^2}{V} \tag{3.13}$$

In a VNE process, a VNR must entirely be mapped (whole population), so in (13), we take all the VNR size. For the SC, when S equals V, it means that every composite in both networks is involved, thereby, K equals the same value as S and V. However, when V is less than S, it implies that only V servers from SN server size are required for the VNE process. Therefore the SC variance is calculated in the following way:

$$\sigma_{SC}^2 = \frac{(FC_{cap} - Imp_{SC})^2}{S-1} \tag{3.14}$$

K equals to $S-1$, which indicates that the VNE process is established with just a sample of servers from SN.

3.3. Problem formulation. In finance, MPT is employed to minimize the risk of investment; therefore, the expected return is maximized. However, in this paper, MPT is presented as a heuristic approach to combine resource importance and standard deviations. Specifically, MPT is established by focusing on one of two goals: First, maximizing the expected return is predicated on a certain risk level. Otherwise, an investor sets a desirable expected return and tries to adjust its associated risk by minimizing the mean assets' standard deviation. The second goal is infeasible due to the certainty of dropping the SCs capacity and an evident constant VCs capacity. Besides, in a VNE approach, it is possible to maximize the substrate resource capacity by reinforcing the DCN with additional high-capacity resources; however, achieving such a heavy task is a costly solution.

The first goal is distinguished as a significantly resource-consuming pattern; it is unreasonable to maximize composite importance. Hence, we consider minimizing the substrate resource capacity by adjusting the Markowitz mean-variance model to suit our MPT-VNE approach. Accordingly, we constructed the composite importance model.

$$
\begin{aligned}
&Find \qquad\qquad OSC = (x_1, ..., x_n), OVC = (y_1, ..., y_m)\\
&So\ as\ to\\
&Min\ [F_{Imp}(x)]\\
&Subject\ to\\
&\qquad\qquad Imp_{SC}(x_i) \geq Imp_{VC}(y_j);\ x \in R^n;\ y \in L^n\\
&\qquad\qquad Imp_{SC}(x_i) \geq 0, Imp_{VC}(y_j) > 0\\
&\qquad\qquad \sum w_{SC} = \sum w_{VC} = 1
\end{aligned}
\tag{3.15}
$$

where:
OSC: an optimal set of SCs | OVC: an optimal set of VCs
L: a set of available VSs | R: a set of available SCs
Imp_{VC}: importance value of a VC | Imp_{SC}: importance value of a SC
$F_{Imp}(x)$: importance minimization function

The optimality of VC mapping lies in the perfect order of every composite in VNR. Thus, our MPT-VNE method seeks to maximize the utilization of local resources in every SC. This yields to a periodic reduction in SCs importance values.

4. Proposed work. Some research has viewed SDN as part of the network virtualization technology landscape due to its heavy use of advanced software solutions. Hence, it will lead to lower expenditure on expensive equipment and centralized management of resources.

4.1. System architecture. The employment of SDN functionalities into cloud infrastructure has elevated DCN management by deploying an efficient resource virtualization process. This latter creates VNs based on resources capacity, availability, and preferred policy of both SPs and end-users.

Accordingly, a real case scenario to our approach would be a SP receiving renting requests of physical resources. By accepting this request, the SP tries to manage the available substrate resources by creating VNs that need to be embedded in the most optimized way to meet his business planes.

From Fig. 4.1, we can highlight four main entities of our design proposal:

1. **Cloud User:** Represent the external source of VNRs that triggers the internal process of VNE.
2. **Service Provider:** Exemplifies the mediator ground between the cloud user and the available cloud services, where it can determine his requirements and make a decision based on the current offer. In our case, this entity can demonstrate the application plane utility thanks to the SDN broker's role of re-ordering the VNRs received from the least-loaded to the most-loaded and then pass them into a verification function to decide if they are apt for further embedding evaluation.
3. **Control Plane:** It has the primary job of maintaining the running services in DCN and informing the SDN broker about free resources. These tasks are managed by the SDN controller, which is typically a server in charge of featuring cloud monitoring, resource control, VN management, and the overall

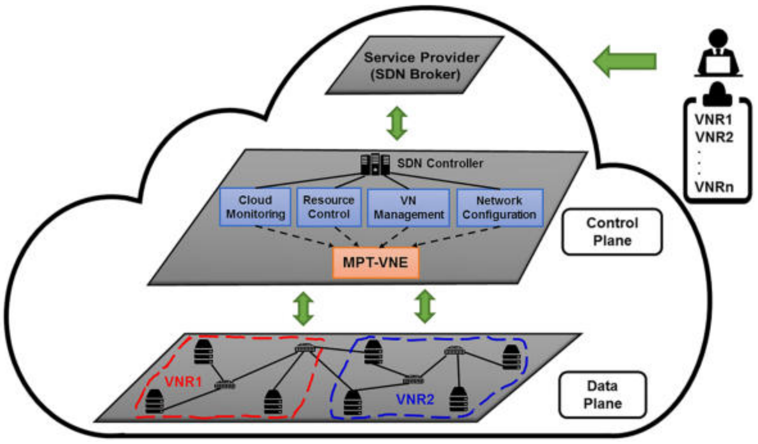

Fig. 4.1: Overview of the proposed SDN-enabled cloud system architecture

network configuration. Furthermore, SDN controller accommodates our core logic designated as MPT-VNE policy by gathering the necessary information in advance like any other heuristic method.

4. **Data Plane (Forwarding Plane):** Contains the underlying types of networking equipment that communicate with the SDN controller by sending network states and forwarding commands to servers. Precisely, to simulate the practicality of this plane, we implemented multiple types of DCNs.

Our system policy applies an offline VNE algorithm (preset VNRs) that depends on the SDN controller to manage the accepted VNRs by the SDN broker in a custom scheduling plan. The initial VNR verification in the application plane does not enable handling the VNRs mapping in real-time, although no remapping procedure is needed afterward.

4.2. MPT-VNE implementation. As VNE techniques rely on VN exigency and the SN's resource availability, our importance-aware policy features the optimal resources in DCNs. Moreover, it determines the most valuable VC in a given VNR (MPT optimizes the correlation pattern between composites). MPT-VNE is originated from two phases, including ranking and mapping.

4.2.1. Ranking of composites. Intuitively, a particular composite rank is calculated based on the node CPU capacity and the combined bandwidth of its attached links. However, since we are dealing with a mapping procedure of VMs, we applied the Eqs. 3.7 and 3.9 to collect the resource requirements that guarantee the running of every VM on a given server.

Algorithm 1 produces four lists of importance and standard deviation for SCs and VSc, which are the essential inputs for ranking and mapping processes. Then, we establish a simple *ranking procedure* to generate two ranked lists $< SC_r >$ and $< VC_r >$. These lists contain the composite ID in conformity with the pre-classified lists of the composites standard deviations.

4.2.2. Mapping Process. In our work, focusing on manipulating the standard deviation of composites capacity is a ruled out objective. Therefore, the standard deviation measurements are only performed for matching purposes between VSs and SCs. We consolidate this matter by engaging the SDN broker to execute a verification function ($VNR\_verify()$) for every VNR. Indeed, this function will return $false$ if the number of VMs is bigger than the number of servers (according to VNE standardization), or the VNR requirements are higher than the DCN capacity.

$$Imp_{SN} = \sum_{i=1}^{N}(Imp_{SC_i}) \tag{4.1}$$

Algorithm 1 Computing of importance and standard deviation

```
    Input: VNR, SN.
    Output: $< Imp\_SC >$, $< Std\_SC >$, $< Imp\_VC >$, $< Std\_VC >$.
 1: for $i \leftarrow < SC > size$ do
 2:     Compute $< Imp\_SC > (i)$;                         ▷ Eqs. 3.4, 3.5, 3.6 and 3.7
 3:     if $< SC > size$ equal $< VC > size$ then
 4:         Compute $< Var\_SC > (i)$;                     ▷ Eq. 3.12
 5:         Compute $< Std\_SC > (i)$;                     ▷ Eq. 3.11
 6:     end if
 7:     if $< SC > size$ greater than $< VC > size$ then
 8:         Compute $< Var\_SC > (i)$;                     ▷ Eq. 3.14
 9:         Compute $< Std\_SC > (i)$;                     ▷ Eq. 3.11
10:     end if
11: end for
12: for $i \leftarrow < VC > size$ do
13:     Compute $< Imp\_VC > (i)$;                         ▷ Eqs. 3.5, 3.8 and 3.9
14:     Compute $< Var\_VC > (i)$;                         ▷ Eq. 3.13
15:     Compute $< Std\_VC > (i)$;                         ▷ Eq. 3.11
16: end for
```

$$Imp_{VN} = \sum_{j=1}^{N}(Imp_{VC_j}) \tag{4.2}$$

The importance of SN and VN are calculated in Eq. 4.1 and 4.2, respectively. A VNR is accepted for further mapping evaluation when the value of Imp_{SN} is higher than Imp_{VN}. A decisive part of our matching policy exists in the mapping process, emphasizing the important values to reinforce the network management.

For every investment operation, MPT provides a logical assets exhibition beginning with the bond-asset and finishing in the equity-asset. With that in mind, in Algorithm 2, we already sorted the composites from lower to higher in terms of standard deviation values, which keeps a coherence affiliation between composites in the matching procedure. Nevertheless, we need also to review the importance values for every composite and proceed with an accurate bandwidth checking. A VNR is scheduled for further processing if the current available bandwidth for SC cannot satisfy the required bandwidth of the compared VC. If the size of $< SC_r >$ is greater than $< VC_r >$ size, we call the *Search_map*() function at line 20, which is a simple iterative function that iterates through $< Imp_{VC} >$ and $< Imp_{SC} >$ to locate the most suitable SC. An optimal SC list $< OP\_SC >$ is returned when the embedding is accomplished.

4.3. General process of MPT-VNE. Typically, every VNR is accompanied by several preferences related to stationary attributes submitted with the list $< workloads >$ in the inputs.

The overall process of MPT-VNE presented in Algorithm 3 is executed through all the SDN layers. Firstly, it is established at the application plane by verifying the received VNRs on line 2. Secondly, it takes place at the control plane and the data plane by encompassing all the previous algorithms while covering the necessary exchange of information. The proposed MPT-VNE is developed by employing a network virtualization strategy (a greedy heuristic method) as a part of the SDN system design. To formalize this work, we acknowledge the following backgrounds:

- We note that our policy does not consider or apply a fault tolerance procedure. Thus, we assume that all the physical equipment in the DCN performs regularly, and no failure can occur, whether at VN or SN levels.
- The integrated matching mechanism enables the node/link mapping function to be treated as a single composite mapping to reduce the execution time.
- We emphasize that our VNE version deals only with static substrate resources in a DCN with redundant VNRs management and does not establish any virtual resource migration.

Algorithm 2 Mapping process

```
    Input: $< SC_r >$ , $< VC_r >$, $< Imp\_SC >$,$< Imp\_VC >$.
    Output: $< OP\_SC >$
1: Sort $< Imp\_SC >$, $< Imp\_VC >$;
2: if $< SC_r >$ equal $< VC_r >$ then
3:     for $i \leftarrow < VC_r > size$ do
4:         if $< Imp\_VC > (i)$ less or equal $< Imp\_SC > (i)$ then
5:             if $< VC_{r_{(BW)}} > (i)$ greater than $< SC_{r_{(BW)}} > (i)$ then
6:                 $Schedule\_map()$;          ▷ Returns false after the timeout has completed
7:                 if $Schedule\_map()$ returns $false$ then
8:                     Print "Mapping failed";
9:                 end if
10:            else
11:                Embed $< VC_r > (i)$ into $< SC_r > (i)$;
12:                Add $< VC_r > (i)$ to $< OP\_SC >$;
13:            end if
14:        else
15:            Print "Mapping failed";
16:        end if
17:    end for
18: end if
19: if $< SC_r > size$ greater than $< VC_r > size$ then
20:     $Search\_map()$;          ▷ Returns false if no fitted SC is found
21:     if $Search\_map()$ returns $false$ then
22:         Print "Mapping failed";
23:     else
24:         Apply instructions from line 3 to 17;
25:     end if
26: end if
27: Return $< OP\_SC >$;
```

Algorithm 3 An overview of the MPT-VNE process

```
    Input: $SN$ , $< VNRs >$, $< workloads >$.
    Output: $< OP\_SC >$
1: for all $VNR \in < VNRs >$ do
2:     if $VNR\_verify()$ equal $true$ then
3:         Generate the required Data using Algorithm 1;
4:         Rank the SCs and VCs and apply the mapping process using Algorithm 2;
5:     else
6:         Print "Mapping failed";
7:     end if
8: end for
```

5. Performance evaluation. In this section, we investigate the performance of our VNE approach through empirical simulation evaluations. These are performed via two types of DCNs, switch-centric DCN (SwCDCNs) and server-centric DCN (SvCDCNs), to provide a comparison between MPT-VNE and other importance-based VNE approaches. Also, we provide a general behavior analysis between the mentioned DCNs.

Table 5.1: Simulation parameters for VNR

VNR design characteristics	**Requirement**
VM Size	100MB – 3 GB
VM RAM	256MB – 4 GB
VM Bandwidth	100MBps – 10 GBps
VM number of CPUs	1 – 4
Link Bandwidth	100MBps – 5 GBps
Number of VMs	4 – 120

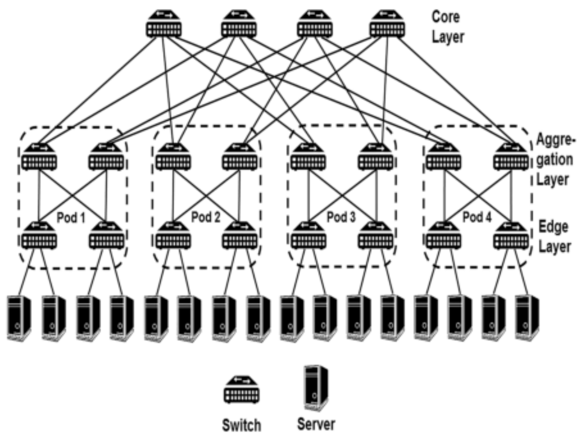

Fig. 5.1: Fat-tree topology with 4 pods

5.1. Simulation setting. In our simulation context, the SDN controller is a logical orchestration entity referred to as a network operating system (NOS) based on the predefined implementation of CloudSimSDN [14]. For modeling SDN-enabled cloud data centers. Throughout the simulations, we used a computer with an Intel Core i5-6200U processor up to 2.8 GHz and 4 GB of RAM.

5.1.1. VNRs setup. Besides the NOS class, CloudSimSDN has two other relevant classes, regarding physical topology generator and virtual topology generator to implement VNRs and DCNs in JSON files. We have generated 23 VNRs where each VNR has been processed for potential mapping in every DCN. A VNR sample contains a list of customized VMs and VLins. Table 5.1 exhibits the possible capacity requirements in the VNRs list, where a VNR can include a minimum of 4 VMs or up to 120 VMs:

We note that once the VNR is generated, an extra output is given regarding its specific workload. The VNR workload will be saved in an excel file and passed in inputs, as shown in Algorithm 3.

5.1.2. DCNs Setup. SwCDCNs rely basically on switches for forwarding the network packets. We focus on the tree-like networks regarding Fat-tree [15], VL2 [16] and Diamond [17], Figs. 5.1, 5.2, and 5.3 exemplify a sample of the implemented SwCDCNs with 16 servers in each.

A SwCDCN instance can contain a minimum of 16 servers or up to 128 servers with a maximum of 0.2 seconds and 12 GB per second of latency and bandwidth, respectively, on the attached links. Further configuration details are available identically in Table 5.2.

In SvCDCNs, servers play the role of a switch (responsible for network routing). They usually consist of programmable servers connected directly with each other and with few low-cost mini-switches. We have selected three SvCDCNs regarding DCell [18], Ficonn [19] and BCube [20]. Figs. 5.4, 5.5, and 5.6 depict a SvCDCNs representation of a level 1 DCell, level 2 Ficonn, and level 1 BCube, respectively. SvCDCNs simulation setup is presented in Table 5.3, where for each instance, we can have up to 20 levels with less maximum latency (0.01 seconds) compared with the SwCDCNs.

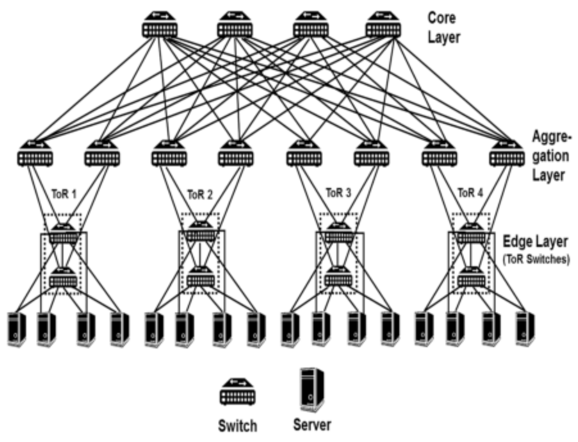

Fig. 5.2: VL2 topology with 4 ToRs

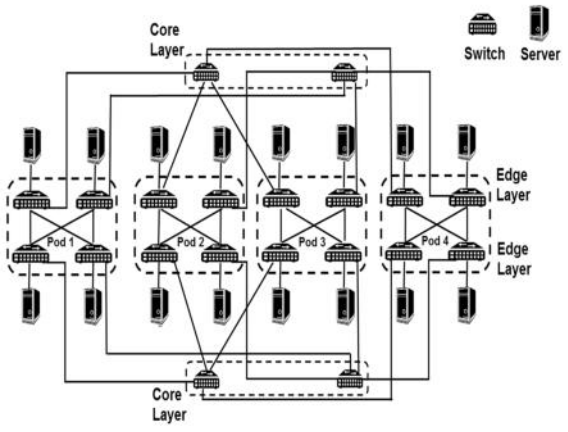

Fig. 5.3: Diamond topology with 4 pods

Table 5.2: Simulation parameters for SwCDCNs

SwCDCN structure characteristics	**Load values**
Number of pods/ ToRs	4 – 8
Number of servers	16 – 128
Number of CPUs	8 – 16
Server RAM	10 – 16 GB
Server storage	10 – 50 GB
Link bandwidth	10 – 12 GB
Link latency	0.008 – 0.2 seconds

We note that depending on the type of DCN; a new DCN instance is implemented of the same DCN based on the number of pods or levels (e.g., we have 2 instances of Fat-tree if we considered to generate it twice, one instance with 4 pods and the second with 8 pods). In our work, the network traffic relies on the forwarding rules generated in the SDN controller, based on the workload size of each VNR. Therefore, we established a logical connection between the SDN controller and the switches in the core layer and all the mini switches in the SwCDCNs and SvCDCNs, respectively.

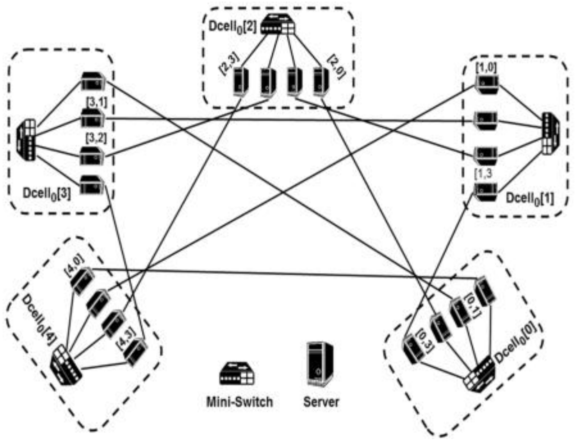

Fig. 5.4: Level 1 DCell topology

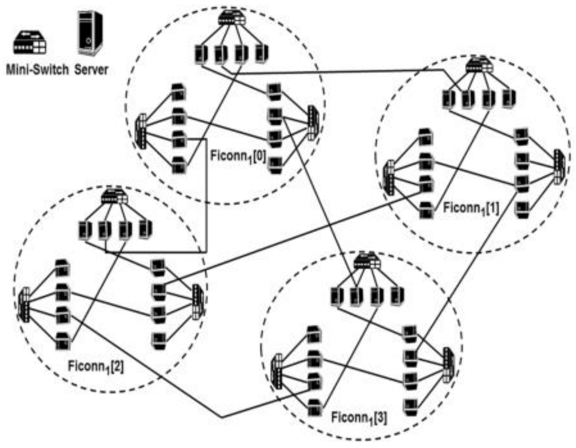

Fig. 5.5: Level 2 Ficonn topology

Table 5.3: Simulation parameters for SvCDCNs

SvCDCN structure characteristics	**Load values**
Number of levels	1 – 20
Number of servers	12 – 128
Number of CPUs	12 – 20
Server RAM	16 – 20 GB
Server storage	10 – 50 GB
Link bandwidth	10 – 16 GB
Link latency	0.001 – 0.01 seconds

5.2. Evaluation methods and metrics. To validate our contribution, we have conducted comparative simulations between the proposed MPT-VNE approach and four similar VNE algorithms, which mainly considers resource capacities as the influencing factor in the mapping process's decisive phase. These approaches are listed in Table 5.4.

In order to keep the comparison in the same context, we modified specific algorithms (NRM, Least-load, and NDC) to assure the embedding is performed in one stage (i.e., mapping VNods and VLins at the same

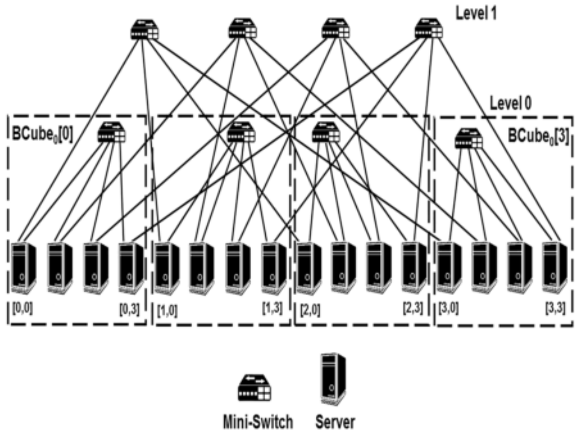

Fig. 5.6: Level 1 BCube topology

Table 5.4: Elected algorithms for results comparison

Importance-based Algorithms	**Description**
RW-BFS-VNE	A one-stage topology-aware heuristicbased on a breadth-first searching using the random walk model to calculate node ranking values [21].
NRM-SP-VNE	A modified node ranking heuristic to support a one-stage embedding using a short path algorithm [22].
Least-SP-VNE	Network method that treats first the nodes with less stress and connectivity merged with the short path algorithm to perform VNE at one coordinated stage. A similar version is already integrated into CloudSimSDN.
NDC-SP-VNE	An enhanced algorithm to determine the node influence by employing the node degree characteristics. Since it is a two-stage VNE algorithm, we added the short path algorithm to preserve the VNE evaluation at the same stage [23].

time) by associating these algorithms with an algorithm that targets the subtract links with more bandwidth capacity. Thereby, VNods and VLins are mapped simultaneously.

To evaluating our work, we focus on comparing the final results with the existing approaches based on the following performance metrics:

1. **Average acceptance ratio:** We denote AR the acceptance ratio of a VNR in a DCN instance, which indicates the accepted VNRs VNR_{apt} from the total received VNR_{rcv} in a given time range t.

$$AR = \lim_{x \to -\infty} \left(\frac{\sum_{t=0}^{T} VNR_{apt}}{\sum_{t=0}^{T} VNR_{rcv}} \right) \tag{5.1}$$

In our VNE model, the accepted VNRs used in AR are already mapped to distinguish them from the initially accepted VNRs in the application plane. The Average acceptance ratio is defined as the following:

$$AR_{avg} = \frac{\sum_{i=1}^{N} AR_i}{\sum_{j=1}^{N} DCN_j} \tag{5.2}$$

Where DCN_j denotes a single DCN instance of the same SN.

2. **Average runtime:** Handled by CloudSimSDN, it implies the execution lifetime of a given approach, including the algorithm mapping processing added to the traffic transmission period.

$$DRun_{avg} = \frac{\sum_{i=1}^{N} DRun_i}{\sum_{j=1}^{N} DCN_j} \tag{5.3}$$

Where $DRun$ is the runtime of a single DCN instance.

3. **Average rate of SNods and SLins utilization:** This metric relies on $< OP\_SC >$ and the selected switches to identify the rate of local resource utilization for every accepted VNR. The utilization rate of nodes NUR and links LUR in a single DCN instance are calculated as follows:

$$NUR = \left(\frac{\sum_{VNR=1}^{N} SNods_{sel}}{Total\_SNods} \right) \times 100 \tag{5.4}$$

$$LUR = \left(\frac{\sum_{VNR=1}^{N} SLins_{sel}}{Total\_SLins} \right) \times 100 \tag{5.5}$$

where $SLins_{sel}$ and $SLins_{sel}$ respectively denote the sum of selected nodes (hosts and switches) and links, which are used repeatedly in a set of accepted VNRs. The average utilization rates are defined as follows:

$$NUR_{avg} = \frac{\sum_{i=1}^{N} NUR_i}{\sum_{j=1}^{N} DCN_j} \tag{5.6}$$

$$LUR_{avg} = \frac{\sum_{i=1}^{N} LUR_i}{\sum_{j=1}^{N} DCN_j} \tag{5.7}$$

5.3. Results analysis. The initial results review revealed a similar performance between the selected DCNs within the same DCN type, due to the primary objective of the methods in comparison (targeting the most critical resources). Therefore, we preferably present results of different DCNs by only their type.

Figure 5.7 illustrates the average ratio of the accepted VNRs relative to a set of VNE methods. We note that regardless of the DCN type, our VNE approach produces a higher acceptance ratio due to the adapted MPT measurement of the composite importance. The MPT-VNE average acceptance ratio is reduced by 22% within the SvCDCNs, resulting from the additional task of traffic forwarding performed by the servers, while this task is only performed by switches in the case of SwCDCNs. Moreover, SvCDCNs generates more traffic density to the server attached links, which will lower the overall acceptance ratio for all the VNE approaches.

By comparing Figs. 5.8 and 5.9, we note that algorithms in comparison within the SvCDCNs took less time to finish the simulation due to their lower structure scalability compared with SwCDCNs that employ more substrate resources (switches and links) for traffic networking.

In both DCN types, we note a similar running time for small VNR sizes in the case of RW-BFS, NDC, and NRM algorithms. RW-BFS algorithm has the most extended runtime owing that to stochastic matrix manipulations. Contrarily, the MPT model employs a low-cost elementary operations in every procedure, including initial calculations, node ranking, and resource matching.

The algorithms' common goal is to greedily choose the most influential node throughout the mapping process. This selection can be observed in Figs. 5.10 and 5.11, where the average node usage is positively correlated with the corresponding average acceptance ratio in the two types of DCNs (i.e., an algorithm with higher AR_{avg} will induce a higher NUR_{avg} and LUR_{avg} while performed in the same comparative conditions). We emphasize that the obtained average utilization values (i.e., summing the NUR_{avg} and LUR_{avg} values) correspond to the most frequently used substrate resources (e.g., the output results of simulating a fat-tree topology with 8 pods, shows that 64 servers, 59 switches, and 103 links are repeatedly selected while mapping

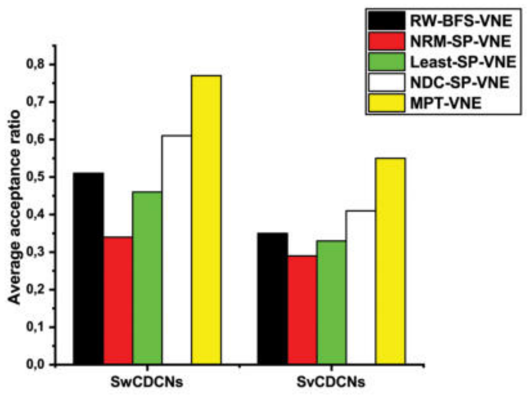

Fig. 5.7: Comparison of accepted VNRs among different DCNs

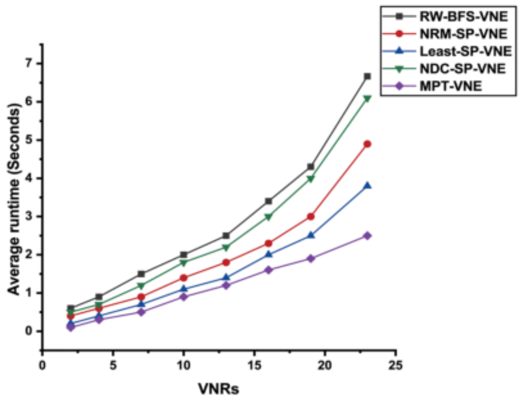

Fig. 5.8: The algorithms runtime for SwCDCNs

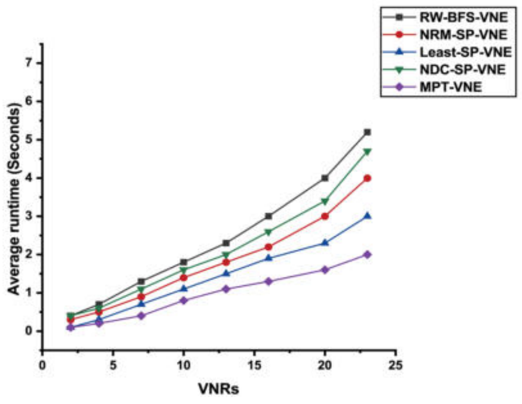

Fig. 5.9: The algorithms runtime for SvCDCNs

12 VNRs from a set of 23 VNRs). The algorithms' performance in SvCDCNs yields a lower average resource utilization than the ones in SwCDCNs. Our proposed algorithm has generated a nearly identical average

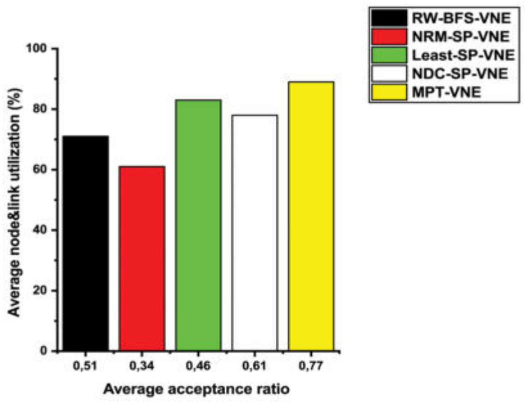

Fig. 5.10: The average node utilization in SwCDCNs

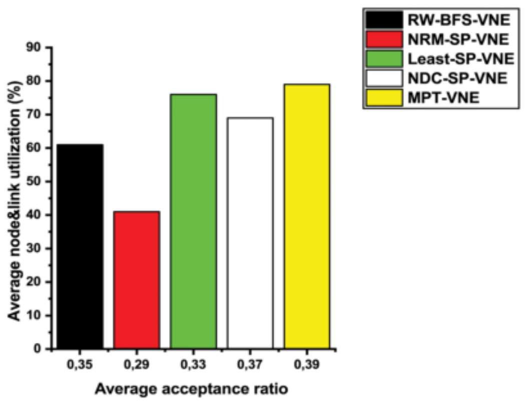

Fig. 5.11: The average node utilization in SvCDCNs

utilization rate with the Least-loaded algorithm (only 3% lower than MPT-VNE depicted in Fig. 5.11). Such an output is generated due to:

- The minimal existence of switches and links.
- The Least-loaded algorithm objective that seeks for the nodes with the lowest capacity and consequently maximizes the overall usage, without forgetting that nodes in SvCDCNs are already loaded nodes (traffic management).

Our approach scored the best average utilization rate thanks to the SC weight factor (latency), which highly increase the same SC selection during the matching process (ranking and mapping). Additionally, higher utilization of the attached links is noted. Nevertheless, for SvCDCNs, we suggest employing enhanced Least-loaded based algorithms in order to get the best possible results.

Notably, Figs. 5.8, 5.9, 5.10 and 5.11 have summarized the aspect of revenue to cost (R/C) relationship. Higher acceptance ratio with less involvement of substrate resources, will increase the SP's profits. Figs 5.10 and 5.11 reveal the average amount of resources repeatedly utilized over the received VNRs. Overall, result analysis proves that SwCDCNs is the most suitable physical structure for importance based VNE techniques.

6. Conclusion. In this paper, we took advantage of this ability to propose an adaptation of financial modeling (Markowitz model) known as modern portfolio theory, intending to provide an improved solution to the VNE problem. By adjusting this mathematical model, MPT proved a sufficient capability to inter-operate with the SDN structure, mainly engaged at the control layer. This work aims to propose a VNE policy that undertakes decision-making through the SDN controller's supervision functionality. MPT-VNE employs the adapted asset allocation of MPT by treating the node and its attached links as a single composite to embed the VNR optimally.

Simulation assessment has focused on examining the proposed approach on two types of implemented DCNs to deeply explore the potential outcomes. Experimental results revealed that our VNE policy outperforms similar algorithms by scoring a higher acceptance ratio and substrate resource utilization with the lowest execution time in both DCNs types. Perspectively, we aim to propose an efficient energy consumption strategy that enables the enhanced effect of MPT-VNE on cloud resources. Besides, we seek to merge the Markowitz model with a scheduling algorithm based on VNR-priority classification to enforce the mapping process via multi-dimensional resource provisions.

REFERENCES

[1] Sasko Ristov, Kiril Cvetkov, and Marjan Gusev. Implementation of a horizontal scalable balancer for dew computing services. *Scalable Computing: Practice and Experience*, 17(2):79–90, 2016.

[2] Bin Wang, Zhengwei Qi, Ruhui Ma, Haibing Guan, and Athanasios V Vasilakos. A survey on data center networking for cloud computing. *Computer Networks*, 91:528–547, 2015.

[3] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, 15(4):1888–1906, 2013.

[4] Hamza Mutaher Alshameri and Pradeep Kumar. An efficient zero-knowledge proof based identification scheme for securing software defined network. *Scalable Computing: Practice and Experience*, 20(1):181–189, 2019.

[5] Harry Markowitz. Portfolio selection, 1959.

[6] Xinbo Liu, Buhong Wang, and Zhixian Yang. Virtual network embedding based on topology potential. *Entropy*, 20(12):941, 2018.

[7] An Song, Wei-Neng Chen, Tianlong Gu, Huaxiang Zhang, and Jun Zhang. A constructive particle swarm optimizer for virtual network embedding. *IEEE Transactions on Network Science and Engineering*, 7(3):1406–1420, 2019.

[8] Zeheng Yang and Yongan Guo. An exact virtual network embedding algorithm based on integer linear programming for virtual network request with location constraint. *China Communications*, 13(8):177–183, 2016.

[9] Bernard M Waxman. Routing of multipoint connections. *IEEE journal on selected areas in communications*, 6(9):1617–1622, 1988.

[10] Koki Inoue, Shin'ichi Arakawa, Satoshi Imai, Toru Katagiri, and Masayuki Murata. Adaptive vne method based on yuragi principle for software defined infrastructure. In *2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR)*, pages 188–193. IEEE, 2016.

[11] Tausifa Jan Saleem and Mohammad Ahsan Chishti. Data analytics in the internet of things: a survey. *Scalable Computing: Practice and Experience*, 20(4):607–630, 2019.

[12] Peiying Zhang, Haipeng Yao, and Yunjie Liu. Virtual network embedding based on the degree and clustering coefficient information. *IEEE Access*, 4:8572–8580, 2016.

[13] Bo Li, Songtao Guo, Yan Wu, and Defang Liu. Construction and resource allocation of cost-efficient clustered virtual network in software defined networks. *Journal of Grid Computing*, 15(4):457–473, 2017.

[14] Jungmin Son, Amir Vahid Dastjerdi, Rodrigo N Calheiros, Xiaohui Ji, Young Yoon, and Rajkumar Buyya. Cloudsimsdn: Modeling and simulation of software-defined cloud data centers. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 475–484. IEEE, 2015.

[15] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. *ACM SIGCOMM computer communication review*, 38(4):63–74, 2008.

[16] Albert Greenberg, James R Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A Maltz, Parveen Patel, and Sudipta Sengupta. Vl2: A scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 51–62, 2009.

[17] Yantao Sun, Jing Chen, Q Lu, and Weiwei Fang. Diamond: An improved fat-tree architecture for large-scale data centers. *Journal of Communications*, 9(1):91–98, 2014.

[18] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. Dcell: a scalable and fault-tolerant network structure for data centers. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, pages 75–86, 2008.

[19] Dan Li, Chuanxiong Guo, Haitao Wu, Kun Tan, Yongguang Zhang, and Songwu Lu. Ficonn: Using backup port for server interconnection in data centers. In *IEEE INFOCOM 2009*, pages 2276–2285. IEEE, 2009.

[20] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. Bcube: a high performance, server-centric network architecture for modular data centers. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 63–74, 2009.

[21] Xiang Cheng, Sen Su, Zhongbao Zhang, Hanchi Wang, Fangchun Yang, Yan Luo, and Jie Wang. Virtual network embedding through topology-aware node ranking. *ACM SIGCOMM Computer Communication Review*, 41(2):38–47, 2011.

[22] Peiying Zhang, Haipeng Yao, and Yunjie Liu. Virtual network embedding based on computing, network, and storage resource constraints. *IEEE Internet of Things Journal*, 5(5):3298–3304, 2017.

[23] Min Feng, Jianxin Liao, Jingyu Wang, Sude Qing, and Qi Qi. Topology-aware virtual network embedding based on multiple characteristics. In *2014 IEEE International Conference on Communications (ICC)*, pages 2956–2962. IEEE, 2014.

Edited by: Katarzyna Wasielewska
Received: May 26, 2022
Accepted: Dec 2, 2022