Achieving Virtual Network Function Load-Balanced Flow Migration in Dynamic Cloud Data Centers

Phillip Aguilera, Christopher Gonzalez, Bin Tang
California State University Dominguez Hills, Carson, USA

Abstract

Virtual Network Functions (VNFs) are software implementation of middleboxes (MBs) (e.g., firewalls) that provide performance and security guarantees for virtual machine (VM) cloud applications. In this paper we study a new flow migration problem in VNF-enabled cloud data centers where the traffic rates of VM flows are constantly changing. Our goal is to minimize the total network traffic (therefore optimizing the network resources such as bandwidth and energy) while considering that VNFs have limited processing capability. We formulate the flow migration problem and design two efficient benefit-based greedy algorithms. The simulations show that our algorithms are effective in reducing the network traffic as well as in achieving load balance among VNFs. In particular, our flow migration algorithms can reduce upto 15% network traffic compared to the case without flow migration.

1. Introduction

Background. Modern cloud data centers adopt virtualization technologies, wherein cloud user applications previously running on multiple physical machines (PMs) can now run as virtual machines (VMs) in a single PM [2]. Recently Network Function Virtualization (NFV) has become an effective new virtualization technique that achieves flexible cloud service management in cloud computing environment [5]. With NFV, proprietary hardware middleboxes (MBs) such as firewalls and cache proxies can now be implemented as virtual network functions (VNFs) running as lightweight containers on commodity hardware [4]. Being provisioned as different services in cloud data centers, VNFs provide security and performance guarantees to cloud user applications in a flexible and cost-effective manner. We refer to the cloud data centers that implement VNFs as *VNF-enabled data centers (VDCs)*.

While the hardware MBs have the dedicated hardware resources such as CPU, memory, or accelerators, software VNFs usually have less packet processing capability and are more prone to software bugs, malfunctions, and misconfiguration. As such, VNFs can be more easily overloaded by the high cloud traffic and fail, causing packet loss and traffic delay in VDCs. Therefore how to load-balance the VNFs is an important problem in VDCs [10].

Fortunately, due to the software implementation of VNFs, it is now possible to replicate and place multiple VNF instances of the same MB type inside the VDCs [3]. As such, the VM traffic just needs to visit one of the instance to achieve the security and performance guarantees brought by VNFs. By distributing VM traffic among multiple VNF instances, it achieves load-balance not only to network traffic thus reducing network congestions, but also to VNFs thus prologing their functional lifetime.

Our Contributions. Recent report about Facebook and other production data centers observes that VM traffic loads including transmission rates and bandwidth demands are highly diverse and dynamic among different user applications [7]. In this paper we identify, formulate and solve a flow migration problem in dynamic VDCs with the goal of minimizing the total network traffic of VM flows. We consider that VNFs have limited processing capability to achieve load-balance of VNFs. We propose two benefit-based efficient heuristics to solve the problem. Via extensive simulations, we show that our algorithms are effective in optimizing the network resources such as bandwidth and energy as well as achieving in load balance among VNFs. In particular, they reduce the network traffic by upto 15% compared to the case without flow migration.

Paper Organization. The rest of the paper is organized as follows. Section 2 gives an overview of the related literature. In Section 3, we formulate the flow migration problem. Section 4 presents the two benefit-based greedy algorithms. In Section 5, we compare the proposed algorithms under different network parameters. Section 6 concludes the paper and discuss possible future work.

2. Related Work

Migrating active in-process flows among VNFs has been an active research in recent years. Due to stateful packet processing in VNFs, some research focuses on guaranteeing loss-free and order-preserving for both flow states and packets. For example, Wang et al. [9] designed a distributed flow migration framework to decouple the state transfer and packets migrations, which allows optimizing the two processes separately and in parallel. Another issue is VNF elasticity control, which studies how to scale out, scale in, and load balance VNFs depending on the traffic load. Sun et al. [8] built a flow migration controller to achieve VNF elasticity control by selecting flows for migrations based upon buffer overflow avoidance and migration cost calculation, with the goal of minimizing the load variance of VNF instances. Qazi et al. [6] proposed to minimize the maximum load of a VNF in order to achieve VNF load-balancing. They show that the problem is NP-hard, thus the optimal load-balancing VNF is time-consuming to compute in a large scale VDCs.

In contrast to all above work, we achieve load-balancing of VNFs by specifying that each VNF has a limited processing capacity. Instead of minimizing the maximum load or the load variance of VNF instances, our goal is to migrate the flows to minimize their total communication cost while respecting the capacity constraint of VNFs.

Our previous work [1] proposed a VNF load-balancing mechanism called LB-MAP. Its goal is to assign VM flows to VNFs to minimize the communication cost of all the VM flows while taking into account of the limited processing capacity of the VNF. However, LB-MAP not only assumes that all VM flows have the same traffic rates but also assumes the traffic rates of VM flows do not change over time. Therefore it fails to consider the dynamic and diverse cloud traffic that commonly exists in cloud data centers. Consequently the algorithms proposed in LB-MAP do not work well for dynamic traffic scenario in cloud data centers. Moreover, instead of considering that all VNFs have the same capacities as in LB-MAP, we consider that different VNFs could have different processing capacities thus targeting a more general scenario.

3. Problem Formulation

System Model. We model a VDC as an undirected general graph $G(V, E)$. $V = V_p \cup V_s$ includes a set of PMs V_p and a set of switches V_s, and E is the set of edges. There are l communicating VM flows $F = \{q_1, q_2, ..., q_l\}$, where flow $q_i = (v_i, v'_i)$ consists of two VMs v_i and v'_i that communicate with each other with some traffic rates. Such rates could be communication frequencies or bandwidth demands of this flow. Let $\mathcal{V} = \{v_1, v'_1, v_2, v'_2, ..., v_l, v'_l\}$, and VM $v \in \mathcal{V}$ is located at PM $s(v) \in V_p$.

Traffic Model. In a dynamic network traffic scenario, the traffic rates of VM flows are constantly changing. Let λ_i denote the traffic rates of q_i at some moment and $\overrightarrow{\lambda} = \langle \lambda_1, \lambda_2, ..., \lambda_l \rangle$ the *traffic rate vector* of the l VM flows at this moment. In Fig. 1, there are two VM flows: $q_1 = (v_1, v'_1)$ and $q_2 = (v_2, v'_2)$ with initial traffic rates $\overrightarrow{\lambda} = \langle 100, 1 \rangle$. As the VM traffic rates change over time in a dynamic VDC, $\overrightarrow{\lambda}$ changes constantly. In Example **??** later, we will show that how dynamic traffic increases the communication costs of VM flows dramatically and how migrating flows from one VNF to another can mitigate the dynamic traffic in a VDC.

VNF Model. There are m VNF instances $M = \{vnf_1, vnf_2, ..., vnf_m\}$ of the same VNF type (i.e., firewalls, load balancers, etc.) in the VDC. We leave the work of traversing multiple VNF types in some order (i.e., service function chains) as future work. We assume that each switch is attached with a server that can install VNFs [11] and all the m VNFs are installed on servers on different switches. Let's assume that vnf_j is installed on switch $w(j) \in V_s$ and $w(j) \neq w(j')$ if $j \neq j'$. For security and performance purposes, each communicating VM flow q_i must traverse one of the VNF instances. Let's assume the capacity of vnf_j, $1 \leq j \leq m$, is κ_j, meaning it can process at most κ_j VM flows at the same time. Obviously we have $\sum_{j=1}^{m} \kappa_j \geq l$; otherwise it cannot find a valid VNF assignment for the VM flows. In Fig. 1, there are two VNF instances: vnf_1 and vnf_2 with $\kappa_j = 1, j = 1, 2$. Table 1 shows all the notations.

Table 1. Notation Summary

Notation	Explanation
V_p	The set of PMs in an VDC
V_s	The set of switches in an VDC
F	The set of l VM flows, $q_i = (v_i, v'_i)$
λ_i	Traffic rate of q_i, $1 \leq i \leq l$
$\overrightarrow{\lambda}$	$\overrightarrow{\lambda} = \langle \lambda_1, \lambda_2, ..., \lambda_l \rangle$ is the traffic rate vector
M	The set of m VNF instances, vnf_j
$s(v)$	The PM where VM v is located
$w(j)$	The switch where vnf_j is installed
κ_j	The processing capacity of vnf_j
$c(u, v)$	The cost between any two nodes u and v
$c_{i,j}$	The communication cost of q_i with vnf_j
μ	The flow migration coefficient
C_c^p	The total comm. cost of all VM pairs at initial VNF assignment p
C_c^f	The total comm. cost after flow migration f
C_m^f	The total flow migr. cost under migration f
C_t^f	The total cost $C_t^f = C_m^f + C_c^f$

Cost Model. Each edge $(u, v) \in E$ has a cost indicating either the delay or energy cost on this edge caused by one unit of VM communication or flow migration. Given any PM or switch u and v, let $c(u, v)$ denote the total cost of all the edges traversed by VM communication or flow migration from u to v. Thus the *communication cost* of any VM flow q_i is $\lambda_i \cdot c\big(s(v_i), s(v'_i)\big)$ and the *flow migration cost* of migrating any VM flow from vnf_i to vnf_j

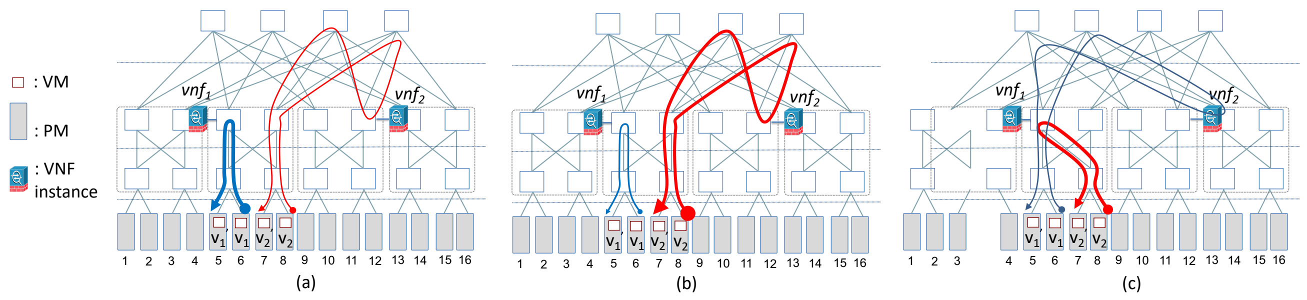

Figure 1. A fat tree-based VDC with 16 PMs. There are two VM flows: (v_1, v_1') and (v_2, v_2') with initial traffic rate vector $\langle 100, 1\rangle$, and two VNF instances: vnf_1 and vnf_2, with capacity $\kappa_j = 1$. (a) The initial VNF assignment is (v_1, v_1') traverses vnf_1 following blue dark line while (v_2, v_2') traverses vnf_2 following red light line, resulting in minimum total cost of $100 \times 4 + 1 \times 8 = 408$. (b) Due to dynamic traffic, the traffic rate vector becomes $\langle 1, 100\rangle$. The resultant VM communication cost becomes $1 \times 4 + 100 \times 8 = 804$, a dramatic increase. (c) By migrating (v_1, v_1') to vnf_2 and (v_2, v_2') to vnf_1, it reduces to $1 \times 8 + 100 \times 4 = 408$. Total flow migration cost is $10 \times 2 = 20$ assuming $\mu = 10$. So total cost of VM migration and communication is $408 + 20 = 428$, a 46.8% decrease compared to the cost before flow migration.

is $\mu \cdot c\big(w(i), w(j)\big)$. Here μ is *flow migration coefficient*, which is the ratio between costs of VM flow migration and VM communication. For example, it could represent the relative size of memory or data packet transferred in VM flow migration and communication. Let $c_{i,j}$ be the communication cost for VM flow q_i when it traverses vnf_j; $c_{i,j} = \lambda_i \cdot \Big(c\big(s(v_i), w(j)\big) + c\big(w(j), s(v_i')\big)\Big)$.

Let $p : [1, 2, ..., l] \rightarrow [1, 2, ..., m]$ denote the *initial VNF assignment*, indicating that $q_i \in F$ is currently traversing $vnf_{p(i)} \in M$ while the capacity constraints of VNFs are satisfied: $|\{1 \le i \le l | p(i) = j\}| \le \kappa_j, 1 \le j \le m$. The communication cost of q_i with p is then $c_{i,p(i)} = \lambda_i \cdot \Big(c\big(s(v_i), w(p(i))\big) + c\big(w(p(i)), s(v_i')\big)\Big)$. The total communication cost of all the l VM flows is $C_c^p \sum_{i=1}^{l} \lambda_i \cdot \Big(c\big(s(v_i), w(p(i))\big) + c\big(w(p(i)), s(v_i')\big)\Big)$.

Note that given the m VNF instances and the l VM flows of different traffic rates, how to assign flows to VNF instances to minimize C_c^p while satisfying the capacity constraints of VNFs has been solved optimally in [1]. However, in a dynamic VDC, as the traffic rate vector $\overrightarrow{\lambda}$ changes, the C_c^p computed in [1] is no longer optimal. In Fig. 1, with initial traffic rate vector $\langle 100, 1\rangle$, the optimal VNF assignment is that (v_1, v_1') traverses vnf_1 following dark blue line while (v_2, v_2') traverses vnf_2 following light red line, resulting in minimum total cost of $100 \times 4 + 1 \times 8 = 408$. Next, due to dynamic traffic, the traffic rate vector changes to $\langle 1, 100\rangle$. The resultant VM communication cost becomes $1 \times 4 + 100 \times 8 = 804$, a dramatic and an almost 100% increase.

Problem Formulation.. Therefore there is a need to migrate the VM flows from one VNF instance to another in order to reduce the network traffic while still satisfying the capacity constraints of VNFs. As migrating flows incurs network traffic and cost, we need to find an optimal flow migration scheme to minimize the total VM flow migration and communication cost. We define a *flow migration function* as $f : [1, 2., ...l] \rightarrow [1, 2, ..., m]$, meaning that the flow q_i will be migrated from its current VNF $vnf_{p(i)}$ to VNF $vnf_{f(i)}$. $f(i) = p(i)$ means the flow of (v_i, v_i') does not migrate. Let $C_m^f = \mu \cdot \sum_{i=1}^{l} c\big(p(i), f(i)\big)$ be the *total migration cost* of all the l VM flows with flow migration f. Let C_c^f be the *total communication cost* of all VM flows *after* VM flow migration f. Let C_t^f be the total cost of VM flow migration and communication after flow migration f. Then, $C_t^f = C_m^f + C_c^f = \mu \cdot \sum_{i=1}^{l} c\big(p(i), f(i)\big) + \sum_{i=1}^{l} \Big(c\big(s(v_i), w(p(i))\big) + c\big(w(p(i)), s(v_i')\big)\Big)$. The objective is to find a flow migration f such that C_t^f is minimized under the processing capacity constraint of VNFs: $|\{1 \le i \le l | f(i) = j\}| \le \kappa_j, 1 \le j \le m$.

Consider the example in Fig. 1 with the new traffic rates $\langle 1, 100\rangle$. Now if we migrate (v_1, v_1') to vnf_2 and (v_2, v_2') to vnf_1, the total communication cost reduces to $1 \times 8 + 100 \times 4 = 408$. The total flow migration cost is $10 \times 2 = 20$ assuming $\mu = 10$. So total VM flow migration and communication cost is $408 + 20 = 428$, a 46.8% decreases compared to the cost before flow migration.

4. Algorithm Solutions

Definition 1: (Benefit of Flow Migration.) Given a VDC graph $G(V, E)$ and flow assignment $p(i)$, indicating that VM flow q_i traverses VNF $vnf_{p(i)}$. The benefit of migrating q_i from $vnf_{p(i)}$ to another VNF vnf_j, denoted as $\mathcal{B}_i^j$, is the total cost reduction resulted from this flow migration; $\mathcal{B}_i^j = c_{i,p(i)} - c_{i,j} - \mu \cdot c(w(p(i)), w(j))$. □

Benefit-based Algorithm 1. Algorithm 1 works as follows.

First, we sort all the VM flows in the non-ascending order of their traffic rates. Then we start with the one with highest traffic rate and migrate it to a VNF that gives the largest benefit without violating this VNF's capacity. This continues until all the VM flows are migrated. Finding the minimum communication cost between of all flows take $O(l \cdot |V|^2 \cdot \log|V|)$. Migrating VM flows to VNF instances takes $O(l \cdot m)$, where m is bounded by $|V|$. Therefore the time complexity is $O(l \cdot |V|^2 \cdot \log|V|)$.

```
Algorithm 1: Benefit-based Algorithm 1.
Input: A VDC $G(V, E)$ with $l$ VM flows and $m$ VNFs
Output: Flow migration scheme $f(i)$, indicating that $q_i$ is
    migrated from $vnf_{p(i)}$ to VNF $vnf_{f(i)}$, and the resulted
    total communication and flow migration cost $C_t^f$.
Notations:
    $i$: the index for VM flows
    $j$: the index for VNF instances
    $p(i)$: $(v_i, v_i')$'s initial assigned VNF $vnf_{p(i)}$
    $f(i)$: $(v_i, v_i')$ migrates to VNF $vnf_{f(i)}$
    $load_j$: the current load of $vnf_j$, initially 0
    $\mathcal{B}_i$: the largest benefit of migrating $q_i$, initially $-\infty$
1.  Compute $C_c^p$, the total communication cost under initial
    VNF assignment $p(i)$;
2.  Sort all VM flows in the non-ascending order of their
    traffic rates. Assume $\lambda_1 \geq \lambda_1, ..., \geq \lambda_l$ WLOG;
3.  for ($i = 1$ to $l$)
4.    $\mathcal{B}_i = -\infty$;
5.    for ($j = 1$ to $m$)
6.      if ($load_j < \kappa_j$)
7.        $\mathcal{B}_i^j = c_{i,p(i)} - c_{i,j} - \mu \cdot c(w(p(i)), w(j))$;
8.        if ($\mathcal{B}_i^j > \mathcal{B}_i$)
9.          $\mathcal{B}_i = \mathcal{B}_i^j$, $f(i) = j$;
10.       end if;
11.     end if;
12.   end for;
13.   $C_c^p = C_c^p - \mathcal{B}_i$;
14.   $load_{f(i)}$++;
15. end for;
16. $C_t^f = C_c^p$;
17. RETURN $\{f(1), f(2), ...f(m)\}$ and $C_t^f$.
```

Benefit-based Algorithm 2. In this algorithm, for VNF instance vnf_j, we migrate κ_j VM flows to vnf_j that have not been migrated such that those migrations give the top κ_j maximum benefits when traversing vnf_j. The running time again is $O(l \cdot |V|^2 \cdot \log|V|)$.

```
Algorithm 2: Benefit-based Algorithm 2.
Input: A VDC $G(V, E)$ with $l$ VM flows and $m$ VNFs
Output: Flow migration scheme $f(i)$.
Notations:
    $i$: the index for VM flows
    $j$: the index for middlebox instances
    $p(i)$: $(v_i, v_i')$'s initial assigned VNF $vnf_{p(i)}$
    $f(i)$: $(v_i, v_i')$ migrates to VNF $vnf_{f(i)}$
    $\mathcal{F}_j$: the set of VM flows migrated to $vnf_j$
    $migrated_i$: true if $q_i$ has already migrated,
      false if not; initially false
1.  Compute $C_c^p$, the total communication cost under initial
    VNF assignment $p(i)$;
2.  for ($j = 1$ to $m$)
3.    $\mathcal{F}_j = \phi$;
4.    for ($i = 1$ to $l$)
5.      if ($migrated_i == false$)
6.        $\mathcal{B}_i^j = c_{i,p(i)} - c_{i,j} - \mu \cdot c(w(p(i)), w(j))$;
7.        $\mathcal{F}_j = \{(i, \mathcal{B}_i^j)\} \cup \mathcal{F}_j$;
8.      end if;
9.    end for;
10.   Sort $\mathcal{F}_j$ in the non-ascending order of $\mathcal{B}_i^j$;
11.   $\mathcal{F}_j = \{(x_1, \mathcal{B}_{x_1}^j), (x_2, \mathcal{B}_{x_2}^j), ...\}$, where $\mathcal{B}_{x_1}^j \geq \mathcal{B}_{x_2}^j ...$;
12.   for ($k = 1$ to $\kappa_j$)
13.     $C_c^p = C_c^p - \mathcal{B}_{x_k}^j$;
14.     $f(x_k) = j$;
15.     $migrated_{x_k} = true$;
16.   end for;
17. end for;
18. $C_t^f = C_c^p$;
19. RETURN $\{f(1), f(2), ...f(m)\}$ and $C_t^f$.
```

5. Performance Evaluation

Simulation Setting. We investigate the performance of the two benefit-based algorithms viz. Algorithms 1 and 2, which are referred as **Benefit1** and **Benefit2**, respectively. We compare them with the case without any flow migration, which is referred to as **NoMigration**. We create a fat-tree VDC with $k = 8$ of 128 PMs, where k is the number of ports each switch has. The VMs are randomly placed on the PMs and the VNF instances are randomly placed on the switches. In all the simulation plots, each data point is an average of 10 runs, and the error bars indicate 95% of confidence interval. For fair comparison, we run the algorithms on the same VDC with the same VM flow and VNF instance placement for each simulation run. In each case, we set the VNF capacity $\kappa_j = \lceil \frac{l}{m} \rceil$.

Varying Number of VM Flows l. Fig. 2 varies the number of VM flows l from 100, 200, ..., to 900 while fixing the number of VNF instances m as 5 and the migration coefficient μ as 100. It shows that the total costs of Benefit1 and Benefit2 are smaller than that of NoMigration, demonstrating that VM flow migration is an effective technique to reduce network traffic. Comparing Benefit1 and Benefit2, as Benefit1 migrates VM flows with high traffic rates first, the resulted benefits of such migrations are generally higher than those obtained in Benefit2, which results in lower total cost for Benefit1.

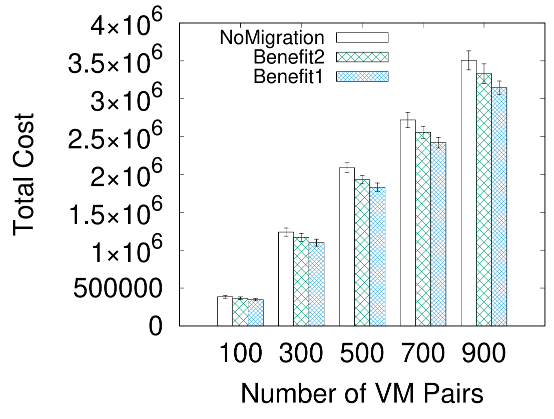

Figure 2. Varying number of VM flows l. $m = 5$ and $\mu = 100$.

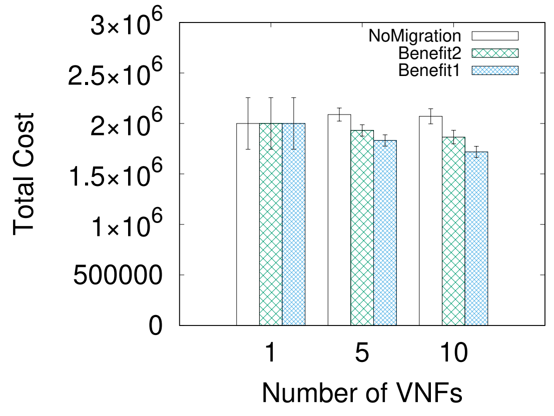

Figure 3. Varying number of VNF intances m. $l = 500$ and $\mu = 100$.

Varying Number of VNF Instances m. Fig. 3 varies the number of VNF instances as 1, 5, 10 while fixing the number of VM flows l as 500 and μ as 100. When $m = 1$, all three yield the same cost as all the VM flows must go through the same and only VNF. With the increase of m, we observe that Benefit1 and Benefit 2 both perform better than NoMigration. In particular, Benefit1 reduces the total cost by up to 10-15%. This again demonstrates that our flow migration problems are valid and our flow migration algorithms are effective.

6. Conclusion and Future Work

We have studied how to migrate VM flows in VNF-enabled cloud data centers to reduce the dynamic traffic while load-balancing VNFs. Our goal is to optimize cloud network resources such as bandwidth and energy consumption. We proposed two efficient flow migration algorithms. Our simulation results show that flow migration is an effective technique to reduce dynamic traffic in cloud data centers. In the future, we will do more simulations to validate our algorithms and check if there exists an optimal and efficient flow migration algorithm.

Acknowledgment

This work was supported by NSF Grant CNS-1911191.

References

[1] M. Alqarni, A. Ing, and B. Tang. Lb-map: Load-balanced middlebox assignment in policy-driven data centers. In *Proc. of IEEE ICCCN 2017*.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, 2010.

[3] A. Jukan F. Carpio, S. Dhahri. Vnf placement with replication for load balancing in nfv networks. In *Proc. of IEEE ICC 2017*.

[4] R. J. Martins, C. B. Both, J. A. Wickboldt, and L. Z. Granville. Virtual network functions migration cost: from identification to prediction. *Computer Networks*, 181(9), 2020.

[5] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba. Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Sur. and Tut.*, 18(1), 2015.

[6] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu. Simplefying middlebox policy enforcement using sdn. In *Proc. of ACM SIGCOMM 2013*.

[7] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren. Inside the social network's (datacenter) network. In *Proc. of SIGCOMM 2015*.

[8] C. Sun, J. Bi, Z. Meng, T. Yang, X. Zhang, and H. Hu. Enabling nfv elasticity control with optimized flow migration. *IEEE Journal on Selected Areas in Communications*, 36(10):2288–2303, 2018.

[9] Y. Wang, G. Xie, Z. Li, P. He, and K. Salamatian. Transparent flow migration for nfv. In *Proc. of the ICNP*, 2016.

[10] C. You and L. M. Li. Efficient load balancing for the vnf deployment with placement constraints. In *Proc. of IEEE ICC*, pages 1–6, 2019.

[11] Y. Zhang, N. Beheshti, L. Beliveau, G. Lefebvre, R. Manghirmalani, R. Mishra, R. Patney, M. Shirazipour, R. Subrahmaniam, C. Truchan, and M. Tatipamula. Steering: A software-defined networking for inline service chaining. In *Proc. of IEEE ICNP 2013*.

Address for correspondence:

Phillip Aguilera
Computer Science Department
California State University Dominguez Hills
Carson, California 90747
paguilera5@toromail.csudh.edu