Cost-efficient VNF Placement Strategy for IoT Networks with Availability Assurance

He Zhu†, Changcheng Huang†
† Department of Systems and Computer Engineering
Carleton University, Ottawa, ON K1S 5B6, Canada
{hzhu, huang}@carleton.ca

***Abstract*—Cloud computing and Network Function Virtualization (NFV) have been leveraged in building Internet-of-Things (IoT) networks because of their ability to reduce cost and to provide elastic and resilient services to rapid-growing IoT devices. In this paper, we propose a novel model to track the impact from placement strategy changes to the cost and the availability of deploying VMs of a Virtual Network Functions (VNF). With our model, the unit resource cost increases as there is less resource left on a host. Putting VMs on the same host takes up more resources of the host but saves network bandwidth cost because of co-location. When enforcing anti-affinity for placing VMs on different hosts, experimental results show increases of both availability and inter-host network bandwidth cost, but varying trends of total costs considering other resources.**

***Keywords*—Placement Policy, Internet of Things, Network Function Virtualization, Cloud Computing.**

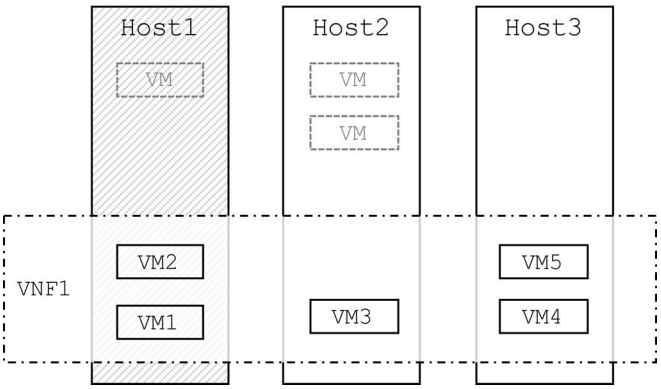

Fig. 1. **A typical scenario to deploy VNF1 with placement policies. Five VMs are deployed in three groups, each placed on a separate host following the inter-group anti-affinity, while having its own VMs placed on the same host as a result of intra-group affinity. A minimum of three VMs are required for VNF1 to be in service. If Host1 is down, VNF1 remains available as there are still three VMs left.**

I. Introduction

Cloud computing delivers virtualized networking services to support large-scale Internet-of-Things (IoT) networks [1]. By leveraging Network Function Virtualization (NFV), which has been a hot topic buzzing among carriers and vendors [2][3], flexible, elastic and cost-efficient Virtual Network Functions (VNFs) are rapidly replacing traditional physical devices [4], benefiting vendors to reduce cost and to adjust services with agility based on fast-changing user demands.

When deploying a VNF in the cloud, the user would always be happy to maximize the availability, while minimizing the cost and maintaining the desired level of security. In a virtual environment, placement rules in a VNF deployment configuration determine on which host each VM is deployed. In practice, they mainly refer to affinity and anti-affinity. A group of VMs with affinity must be deployed on the same host. On the contrary, anti-affinity of a group of VMs ensures that each VM in the group is deployed on a different host. A scenario deploying a VNF with placement policies enforced is shown in Fig. 1.

The affinity rule helps reduce communication costs between VMs within the VNF, because VMs on the same host connect to each other using virtual networks, which require no physical network infrastructure. The confidentiality of the VNF also has less chance to be breached as there is no internal data being passed around between VMs, which may contain vulnerable logging or debugging information. However, for IoT networks, the hosts may not be highly reliable and may face outage due to hardware and environmental constraints. To address such issues, anti-affinity rules are used for high availability (HA) uses in case one host is down. As a trade-off, anti-affinity would not bring the benefits of low communication costs and higher confidentiality.

In this paper, our interest is to find effective placement policies for each configuration of VNF, and to achieve lower costs, while still satisfying the availability and confidentiality requirements. Our contributions are:

(i) A cost model factoring in host failure and inter-host traffic.
(ii) A mixed integer linear programming to minimize the cost based on our cost model, while maintaining the availability requirements.
(iii) Numerical results showing the effectiveness and efficiency of the algorithm.

We divide the contents into the following sections. The related work is illustrated in Section II. Section III formulates the problem. Then the experimental results are shown in Section IV. Section V concludes the paper.

II. Related Work

NFV has been well utilized in IoT networks [5] [6]. Challenges in IoT was highlighted in [5], following a proposed

multi-layered IoT architecture involving NFV. NFV-enabled edge nodes were leveraged to have VMs deployed connecting IoT gateways. Recently, the concept of edge computing was presented [7] for providing services with lower latency at the network edge for IoT devices. In edge computing, NFV is actively applied for adapting dynamic changes of the networks and connected IoT devices.

For the VM placement issue in a cloud data center, a comprehensive study of the VM placement and consolidation techniques used in cloud was presented in [8]. The VM placement problem and various approaches were reviewed. The placement techniques were classified as constraint programming, bin packing, stochastic integer programming and genetic algorithm. With an optimal technique, [9] aimed to minimize the number of required VMs, with considerations of each VMs resource limitation. A VM placement algorithm was proposed in [10] with the objective to minimize carbon footprint. Clayman et al. [11] described an architecture based on an orchestrator that enabled automated placement of virtual network and processing resources across the physical resources of the network. We have presented a method [12] to evaluate vulnerability of applications offloaded to the cloud that can help estimate security concerns of cloud-based VNFs.

Existing work does not appear to consider the combination of low cost and high availability together with practical placement policies, which are affinity and anti-affinity. In this paper, these factors are weighed in, and the strategy is ready for use by real-world NFV scenarios.

III. Problem Formulation

Suppose a VNF has an elastic group of VMs to be deployed on a Virtual Infrastructure Manager (VIM). The VMs can be deployed on any of the hosts available from the VIM. We model the VNF deployment as a weighted directed graph, denoted by $G = (V, L)$, where V is the set of all VMs of the VNF, and L is the set of links between each two VMs exchanging data. These links form the internal VNF network, dedicated to transmitting data for inter-VM coordination.

A. Higher Availability With VM Redundancy

Let the minimum number of VMs required by the VNF denoted by n_m. That being said, if the number of available VMs is at least n_m, the VNF is then considered in service. If the number of VMs available is reduced to less than n_m, the VNF is deemed down.

An intuitive way to increase the availability of the VNF is redundancy. Scaling out a VNF is enabled in form of deploying extra VMs, so that even if some VMs are down, there are still more than n_m VMs in service. The total number of VMs in the VNF, denoted by n_V, must satisfy that $n_V > n_m \geq 0$.

There are two scenarios that VMs are unavailable. We first use S to denote the event that a VM is out of service due to its own issues, most likely root-caused by the software running on the VM, rather than a host failure. Based on its definition, S on one VM is independent from those on other VMs. The probability of S for each VM, denoted by p_S, is assumed to be equal as all VMs are running software identical to each other. We then factor in the availability of the hosts: if one host is down, all VMs deployed on it would be out of service. We use W to represent the event that one VM is out of service as a result of host failure. For any VM, W is independent from S. In comparison with S, events W for VMs deployed on the same host are correlated. The probability of W is denoted by p_W. Because of the similarity of the hosts within the same VIM, we assume that p_W is the same for each host.

Let the set of all available hosts denoted by H. The total number of hosts in H is denoted by n_H. Let x be the actual total number of VMs in service on all hosts. Considering the events of S and W, x can be less than the total VMs deployed n_V. To satisfy the minimum number of VMs deployed, there must be $n_V = |V| = \sum_{i=1}^{n_h} |V_i| \geq n_m$.

Since events S on different VMs are independent from each other, we use *binomial distribution* to describe the availability with a certain number of VMs factoring S alone. Out of n_V VMs deployed, the probability of having x VMs in service is $\binom{n_V}{x}(1-p_S)^x p_S^{n_V-x}$. As x cannot be less than n_m, the probability of having at least n_m VMs in service is

$$\sum_{x=n_m}^{n_V} \binom{n_V}{x}(1-p_S)^x p_S^{n_V-x} \tag{1}$$

We further factor in the event W. Let y be the number of hosts in service out of all n_H hosts, such that $0 \leq y \leq n_H$. The probability of having y hosts available is

$$\binom{n_H}{y}(1-p_W)^y p_W^{n_H-y} \tag{2}$$

With only y hosts available, there still needs to be at least n_m VMs in service. Suppose each VM has the same probability to be deployed on any of the n_H hosts, which means each host has the probability of $1/n_H$ to deploy each VM. Therefore, the probability of at least n_m VMs in service on the condition that there are y hosts available is

$$\binom{n_H}{y}(1-p_W)^y p_W^{n_H-y} \sum_{x=n_m}^{n_V} \left(\frac{y}{n_H}\right)^x \tag{3}$$

As S is independent from W, the probability of having at least n_m VMs in service with the consideration of host availability is shown in Eqn. (4). If the minimum probability of the VNF staying in service is p_a, then the probability below must not be less than p_a:

$$\sum_{y=1}^{n_H} \sum_{x=n_m}^{n_V} \binom{n_H}{y}(1-p_W)^y p_W^{n_H-y} \binom{n_V}{x}(1-p_S)^x p_S^{n_V-x} \left(\frac{y}{n_H}\right)^x \geq p_a \tag{4}$$

The equation above reveals that the risk from host availability is diluted by introducing multiple hosts. By deploying redundant VMs on different hosts, we are able to relax the requirement for p_S and p_W, i.e., using cheaper commodity services to replace high-end yet much-more-expensive servers.

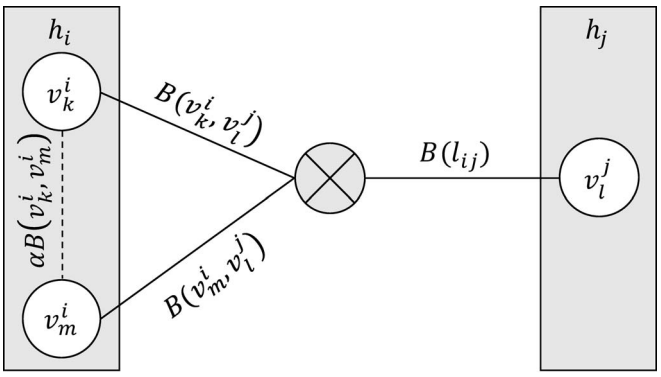

Fig. 2. **Inter-host network bandwidth cost v.s. Intra-host network bandwidth cost converted to CPU cost.**

B. Inter-host Network Bandwidth Cost Due to Redundancy

Redundancy and higher availability come at a cost: extra resources are used for hosting redundant VMs, and extra traffic occurs between VMs working together. Let i be the index of a host, where $i = 1, 2, 3, , n_H$. Also let the ith host in H denoted by h_i. We further denote the kth node on the ith host as v_k^i. Let $h(v_k^i)$ be the function returning which host v_k^i is deployed on. We divide V into multiple subsets by $h(v_k^i)$: $V = \sum_{i=1}^{n_h} V_i, \forall v_k^i \in V_i, \exists h(v_k^i) = h_i$.

Consider v_k^i on Host h_i sends traffic to v_l^j on Host h_j. Let $B(v_k^i, v_l^j)$ be the total bandwidth (BW) used from v_k^i to v_l^j. The required traffic throughput between two hosts must not exceed the designed bandwidth for the inter-host network. We denote the network link between Host h_i and h_j as l_{ij}. Each link has an associated total bandwidth $B(l_{ij})$. Let the residue bandwidth of l_{ij} denoted by $R_B(l_{ij})$. We calculate the residue bandwidth of the link as following:

$$\begin{aligned} R_B(l_{ij}) = B(l_{ij}) &- \sum_{v_k^i \in V_i, v_l^j \in V_j, j \neq i} B(v_k^i, v_l^j) \\ &- \sum_{v_k^i \in V_i, v_l^j \in V_j, j \neq i} B(v_l^j, v_k^i) \end{aligned} \tag{5}$$

C. CPU Cost Due to Redundancy

Despite of the inter-host network bandwidth cost, splitting the workload across redundant VMs consumes extra CPU resources. There are two aspects contributing to the CPU consumption: CPU cycles used for coordinating with other VMs, and intra-host communication between VMs. We assume that the CPU usage by a VM is proportional to its total network traffic load, including both traffic sent and received. Let the conversion ratio from the unit network bandwidth usage to the unit CPU usage denoted by α. Also let the conversion ratio from the intra-host unit network bandwidth usage to the unit CPU usage denoted by β. For a VM v_k^i deployed on Host h_i, all its network traffic accounts for its CPU cost with the ratio α. Plus, the intra-host portion of the traffic accounts for an extra CPU cost with ratio β.

We define $C(h_i)$ as the total CPU capabilities of Host h_i. Similar to the residual network bandwidth, let the residual CPU capabilities of h_i denoted by $R_c(h_i)$. We have

$$\begin{aligned} R_C(h_i) = C(h_i) &- \beta \sum_{v_k^i, v_l^i \in V_i, k \neq l} B(v_k^i, v_l^i) \\ &- \alpha \sum_{v_k^i \in V_i, v_l \in V, k \neq l} B(v_k^i, v_l) \\ &- \alpha \sum_{v_k^i \in V_i, v_l \in V, k \neq l} B(v_k^i, v_l) \end{aligned} \tag{6}$$

D. Mixed Integer Linear Programming Formulation

The owner of the VNF typically aims to minimize the cost when operating cloud-based services. We provide three sets of cost weights in order to combine various types of resources involved in deploying the VNF together. Specifically, A_i^N is the cost weight assigned to Host h_i for hosting a new VM. Suppose Host h_i can deploy a maximum number of $M(h_i)$ VMs. The cost of hosting a new VM for a host is proportional to its occupancy ratio. It will reach its maximum value of A_i^N when the host is fully occupied. In other words, the more occupied a host becomes, the pricier it will become. A_{ij}^B is the cost weight assigned to the network usage between Host h_i and h_j. The cost of the network is proportional to its normalized bandwidth usage, with its maximum value A_{ij}^B when the bandwidth of the network is fully utilized. Let A_i^C stand for the cost weight assigned to Host h_i for its CPU usage. Like the other two cost weights, it is proportional to the CPU usage. It will reach the maximum value A_i^C when the CPU usage of the host is maxed out. The problem is formulated as a mixed integer linear programming (MILP).

Variables

- $|V_i|$: A variable denoting the number of VMs deployed on Host h_i. $|V_i|$ can only be integers. The total number of VMs for the VNF n_V can be described as:

$$n_V = |V| = \sum_{i=1}^{n_H} |V_i| \tag{7}$$

- $B(v_k^i, v_l^j)$: A variable denoting the bandwidth consumed by traffic from v_k^i to v_l^j.

Objective

$$\begin{aligned} Minimize \sum_{h_i \in H} \frac{A_i^N |V_i|}{M(h_i)} &+ \sum_{h_i, h_j \in H, i \neq j} \frac{A_{ij}^B}{R_B(l_{ij}) + \delta} \\ &+ \sum_{h_i \in H} \frac{A_i^C}{R_C(h_i) + \delta} \end{aligned} \tag{8}$$

Constraints

$$B(l_{ij}) \geq \sum_{v_k^i \in V_i, v_l^j \in V_j, j \neq i} \Big[B(v_k^i, v_l^j) + B(v_l^j, v_k^i) \Big] \tag{9}$$

$$\sum_{y=1}^{n_H}\sum_{x=n_m}^{n_V}\binom{n_H}{y}(1-p_W)^y p_W^{n_H-y} \binom{n_V}{x}(1-p_S)^x p_S^{n_V-x}\left(\frac{y}{n_H}\right)^x \geq p_a \tag{10}$$

$$\begin{aligned} C(h_i) \geq & \beta \sum_{v_k^i, v_l^i \in V_i, k\neq l} B(v_k^i, v_l^i) \\ & + \alpha \sum_{v_k^i \in V_i, v_l \in V, k\neq l} B(v_k^i, v_l) \\ & + \alpha \sum_{v_k^i \in V_i, v_l \in V, k\neq l} B(v_k^i, v_l) \end{aligned} \tag{11}$$

Remarks

- Function (8) is the objective function. It tries to minimize the total cost, while keeping the availability above the minimum level required. In the objective function, there is a small positive number δ to avoid dividing by zero. Both the network and the CPU load are normalized by their respective residual capacity. Therefore, the host placement policy tends to find a sweet point minimizing the cost by using less hosts, while not exhausting them.
- Constraint (9) is the link bandwidth capacity bound between each two hosts. Traffic transmitted between any two hosts h_i and h_j must not exceed the corresponding bandwidth capacity $B(l_{ij})$.
- Constraint (10) sets the bottom line of the VNF availability, i.e., the probability of at least n_m VMs in service must be greater than or equal to p_a.
- Constraint (11) is the CPU capacity bound for each host. The CPU used by VMs coordinating with each other and by intra-host communications must not exceed the CPU capacity of the host $C(h_i)$.

IV. Numerical Results

In this section, we illustrate the numerical results of the cost changes for: (1) various types of VNFs under the same cloud host pricing period, each with their unique traffic and CPU load patterns. We are interested in finding out the impact to VNF placement strategy from its cloud usage profile. (2) Same type of VNF under different cloud host pricing period. We are curious if it is beneficial for the placement policy to be adjusted dynamically upon a pricing change.

A. Assumptions

To clearly demonstrate the trends we focus on, the following assumptions are made to simplify the modeling of the problem without losing the generality.

1) Only one type of Network Function (NF) is used. 2) The unit costs of the CPU and network bandwidth for all hosts are the same. We denote the unit bandwidth cost by $A(b)$. 3) All VMs have the same amount of CPU and network bandwidth usage. This can be achieved by load balancing. The CPU usage of each VM is fixed. The network bandwidth usage of each VM is variable. We denote the bandwidth consumed between an two VMs by b, such that $B(v_k^i, v_l^j) = b, \forall v_k^i \in V_i, v_l^j \in V_j, j \neq i$. 4) All hosts and links between them have equal amount of CPU and network bandwidth resources. The CPU resource of each host is fixed. The network bandwidth resource of each host is variable. We denote the total bandwidth of each host by B, such that $B(h_i) = B, \forall h_i \in H$.

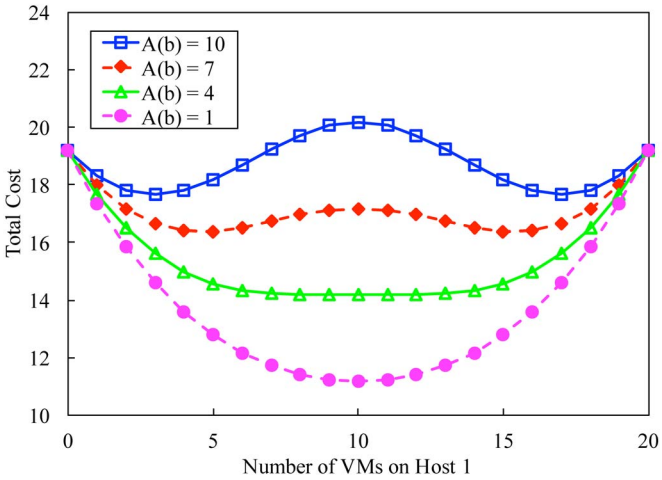

Fig. 3. **Comparison of total costs with 4 different unit bandwidth costs, denoted by $A(b)$. From the results, it is obvious that lower values of $A(b)$ have resulted in more equally loaded hosts. With lower values of $A(b)$, CPU cost would dominate the total cost given that the bandwidth cost is small, and evenly-distributed VMs have reduced total CPU cost.**

B. Two-host Placement Scenario

We start from two-host VNF placement scenario, which is the simplest version of introducing multi-host deployment for a VNF. The two hosts involved are denoted by Host 1 and Host 2, respectively. In the two-host scenario, we choose a deployment with 20 VMs and track the costs of varying numbers of VMs deployed on Host 1.

1) Impact of Unit Bandwidth Cost: The first group of experiments attempts to reveal the impact of unit bandwidth cost $A(b)$. In the two-host scenario, The unit CPU usage cost, CPU usage of each VM, and bandwidth usage between each two VMs all stay the same. The only variable is $A(b)$. Fig. 3 demonstrates the total cost changing trends with four different values of $A(b)$.

The results indicate that changes to $A(b)$ does affect placement decisions. When $A(b)$ is small, it is worthwhile to reduce CPU load and cost for both hosts by distributing VMs to both hosts with the trade-off of increasing bandwidth cost which is less than CPU cost. When $A(b) = 1$, the total cost is the lowest when there are 10 VMs on each host. When $A(b)$ gets higher, placing many VMs on a different host would cause too much extra bandwidth cost. Therefore, finding the sweet point becomes the key to the minimum total cost. In comparison, when $A(b) = 10$, keeping 4 VMs on one host and 16 on the other is the optimal placement decision.

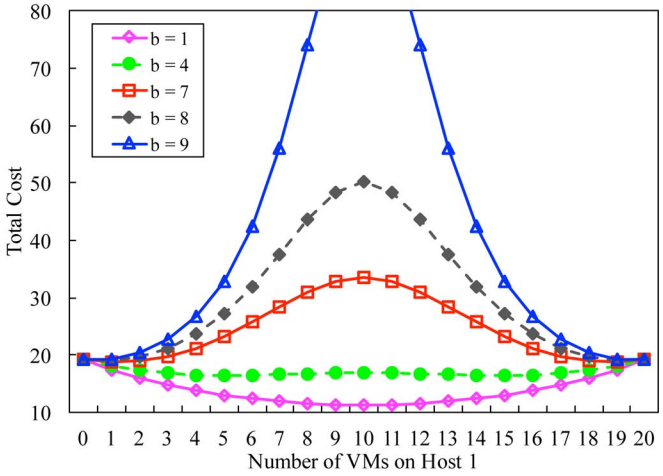

Fig. 4. **Total costs changes caused by 5 unit bandwidth costs per VM, denoted by b. Lower values of b lead to more evenly placement decisions for the optimal total cost.**

2) Impact of VM Bandwidth Usage: Another group of results show how the network bandwidth usage of each VM, b, can impact the placement decision. Fig. 4 tells the fact that the total cost increases rapidly with larger values of b if VMs are placed on the two hosts on a half-and-half basis. Our placement strategy tends to put less VMs on a second host with high VM bandwidth usage. When $b = 1$, which means there is a small bandwidth usage for each VM, the total cost reaches its lowest point by distributing the VMs evenly on two hosts. Under a much bigger VM bandwidth usage, $b = 9$, the same placement strategy would result in sky-rocketed total cost. Instead, putting one VM to a second host would make the lowest cost, while increasing the VNF availability.

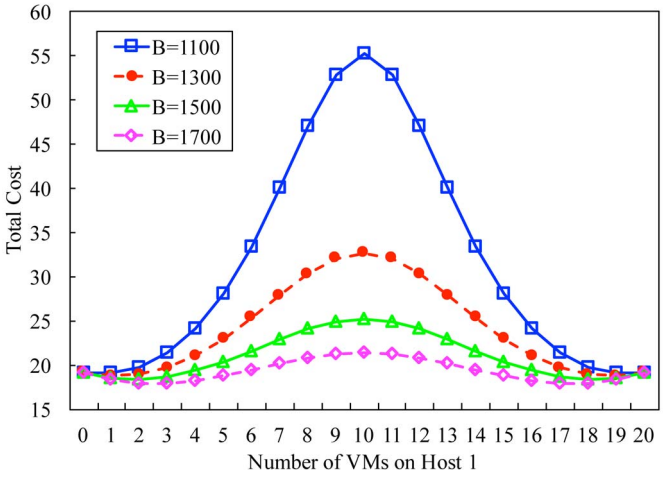

Fig. 5. **Higher total bandwidth available for a host, denoted by B, can attract more VMs deployed on it. Higher residue bandwidth for all hosts will make the model less sensitive to the bandwidth cost.**

3) Impact of Host Total Bandwidth: The total bandwidth of the host B is also a factor of making the placement decision. In the third group of experiments, we assign various amounts of total bandwidth to each host and watch the total cost changes. From Fig. 5, we learn that deploying VMs on a host with more total bandwidth makes significantly less total cost compared to a deployment on a bandwidth-tight host. With $B = 1700$, the total cost is insensitive to the VM placement changes, encouraging placing VMs on different hosts to increase availability. But with a tighter total bandwidth, $B = 1100$, the total cost becomes more affected by VM placement changes, discouraging VMs to be placed at various locations.

V. Conclusion

In this paper, we have presented a VNF placement strategy for IoT networks that promotes reducing the cost by introducing extra hosts and balancing the workload of the hosts. By leveraging affinity and anti-affinity placement rules, VMs can be moved from an overloaded, expensive host to a less busy one to achieve lower cost. Meanwhile, the availability of the VNF has also been profoundly improved as a result of multiple hosts. Our future work is to consider VNFs with combinations of different NFs and with service chaining.

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[2] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "Clickos and the art of network function virtualization," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2014, pp. 459–473.

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[4] Statista, "NFV and SDN market size worldwide by region from 2015 to 2019," https://www.statista.com/statistics/461573/sdn-and-nfv-markets-worldwide-by-region/, [Online; accessed Oct-27-2016].

[5] N. Omnes, M. Bouillon, G. Fromentoux, and O. Le Grand, "A programmable and virtualized network & it infrastructure for the internet of things: How can nfv & sdn help for facing the upcoming challenges," in *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*. IEEE, 2015, pp. 64–69.

[6] R. Vilalta, A. Mayoral, D. Pubill, R. Casellas, R. Martínez, J. Serra, C. Verikoukis, and R. Muñoz, "End-to-end sdn orchestration of iot services using an sdn/nfv-enabled edge node," in *Optical Fiber Communications Conference and Exhibition (OFC), 2016*. IEEE, 2016, pp. 1–3.

[7] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan 2017.

[8] Z. Usmani and S. Singh, "A survey of virtual machine placement techniques in a cloud data center," *Procedia Computer Science*, vol. 78, pp. 491 – 498, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916000958

[9] U. Bellur, C. S. Rao, and S. D. M. Kumar, "Optimal placement algorithms for virtual machines," *CoRR*, vol. abs/1011.5064, 2010. [Online]. Available: http://arxiv.org/abs/1011.5064

[10] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Euro-Par 2013 Parallel Processing - 19th International Conference, Aachen, Germany, August 26-30, 2013. Proceedings*, 2013, pp. 317–328.

[11] S. Clayman, E. Maini, A. Galis, A. Manzalini, and N. Mazzocca, "The dynamic placement of virtual network functions," in *2014 IEEE Network Operations and Management Symposium, NOMS 2014, Krakow, Poland, May 5-9, 2014*, 2014, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/NOMS.2014.6838412

[12] H. Zhu, C. Huang, and J. Yan, "Vulnerability evaluation for securely offloading mobile apps in the cloud," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, Nov 2013, pp. 108–116.