



# Automated Multi-Swarm Networking with Open Baton NFV MANO Framework

Jun-Sik Shin<sup>1(✉)</sup>, Mathias Santos de Brito<sup>2,3</sup>, Thomas Magedanz<sup>2</sup>,  
and JongWon Kim<sup>1(✉)</sup>

<sup>1</sup> EECS, Gwangju Institute of Science and Technology (GIST),  
Gwangju, South Korea

{jsshin, jongwon}@nm.gist.ac.kr

<sup>2</sup> NGNI, Fraunhofer FOKUS, Berlin, Germany

{mathias.santos.de.brito,

thomas.magedanz}@fokus.fraunhofer.de

<sup>3</sup> DCET, Universidade Estadual de Santa Cruz, Ilhéus, Brazil

**Abstract.** Container-based Network Functions Virtualization (NFV) and multi-site/multi-cluster service orchestration are a critical topic in the field of ICT infrastructure. Academia, Industry and Open Source projects are actively working on the technology. With the trends, Open Baton, an implementation of the ETSI NFV MANO Reference Architecture, started efforts to orchestrate network services over multiple Docker Swarm clusters. To achieve that, Open Baton would require an additional feature to configure an overlay networking over multiple swarm clusters, since Docker Swarm does not support multi-cluster service. In this paper, we discuss our design and implementation of the Multi-Swarm Networking Helper in Open Baton, which configures an L2 overlay networking over multiple Docker Swarm clusters by leveraging on a third-party Docker networking driver.

**Keywords:** Multi-cluster networking · Container networking  
Service orchestration with NFV MANO

## 1 Introduction

Container-based NFV (Network Functions Virtualization) is a topic of interest in the field of ICT infrastructure. NFV researches previously focused on bare-metal-based PNF (Physical Network Functions) to address performance issues in VM (Virtual Machine)-based VNF (Virtual Network Functions). However, PNF occupies much hardware resources, and it is difficult to isolate multiple of them in a single box. Therefore, an NFV research line started to focus on Containerized Network Function (CNF), because of advantages of this technology, such as scalability, agility and resource efficiency, delivering a performance equivalent or near to the performance observed when using bare-metal.

Network service orchestration is another popular topic in NFV domain. According to the definition of the European Telecommunications Standard Institute (ETSI), network service orchestration is lifecycle management (deployment, update and remove)

of a network service, a composition of network functions such as firewall and load balancer, over resource cluster(s) [1]. Single-cluster service orchestration based on Virtual Machines matured with the efforts of researchers and open source communities, a natural step forward is to extend it to multi-site/multi-cluster service orchestration based on containers, which is currently an active research topic.

In this paper, we demonstrate an Open Baton NFV MANO (MANagement and Orchestration) framework [2] to orchestrate CNF-based multi-cluster service over Docker Swarm. We extended the Open Baton Docker Swarm VIM/VNFM, under development at Fraunhofer FOKUS, to allow it to orchestrate multi-cluster services over Docker Swarm clusters. In our scenario, the CNFs of a given service should work on an L2 overlay network regardless of the type of the service (single-cluster or multi-cluster). However, the default networking of Docker Swarm has not enough features to support multi-cluster services. Therefore, it is required an additional networking feature for Open Baton to allow the configuration of multi-Swarm Networking, as an overlay network created over multiple Docker Swarm clusters.

In this paper, we present the Multi-Swarm Networking Helper, an additional feature compliant to Open Baton MANO Framework. The feature configures multi-Swarm networking by leveraging on the Weave Net driver, a third-party Docker networking plugin, during the deployment phase of a network service in Open Baton. We then evaluate the functional aspects of the implementation in a real multi-site testbed.

## 2 Backgrounds and State-of-the-Art

Since the introduction of the concept of container-based NFV, A research line claims that container is more appropriate than a virtual machine for deploying network functions. [3, 4] showed CNF has higher performances concerning resource utilization, agility, and scalability compared to VNF. With the advantage of CNF, the authors of [4] suggested Glasgow Network Function that is a container-based NFV platform targeted for orchestrating Linux Container (LXC)-based CNFs over resource-constrained edge boxes. Even though Glasgow Network Function made great progress on CNF orchestration, the design was hard to be aligned with OpenBaton. At the same time, researchers are also focusing on multi-site VNF orchestration. [5–7] insisted that previous NFV studies not show proper solutions to support multi-domain/multi-site VNFs orchestration. The authors suggested their own multi-domain VNFs orchestration framework. However, the works are not appropriate to be applied for CNF orchestration, because the designs do not consider addressing different characteristics of CNF orchestration. The multi-site/multi-cluster orchestration of CNFs combining two domains has still challenged regarding implementation.

OpenStack project, a popular open source cloud operating system, has developed its subprojects, Tacker and Tricircle, for supporting multi-site VNF orchestration. OpenStack Tacker acting as NFVO as well as VNFM orchestrates VNFs over multiple OpenStack clusters. In the other hands, OpenStack Tricircle can configure service (tenant)-level L2/L3 overlay networks over multi-site OpenStack clusters. Combining two projects could orchestrate VNF-based multi-site services, but it has not supported container-based service orchestration yet. A proposal about Kubernetes VIM plugin is

actively being discussed for OpenStack Tacker to be able to orchestrate multi-site services consisting of CNFs as well as VNFs. Meanwhile, Docker Swarm and Kubernetes approach a different way to coordination of multi-site clusters. They federate multiple clusters at the level of Identity and API, but container networking over multiple clusters is not focused.

With trends on containerized NFV, the Open Baton community developed Docker VIM (Virtual Infrastructure Manager), and VNFM (Virtual Network Function Manager) to support Docker container-based infrastructures. Open Baton is an open source NFV MANO framework developed and supported by Fraunhofer FOKUS. It is a reference implementation of ETSI MANO specification [8]. However, Docker VIM/VNFM should manage each of Docker-enabled boxes independently. It increases the management complexity of the resources and makes Open Baton user need to be aware of too much detail of the underlay resources for deploying a service. To resolve this issue, Open Baton could leverage on a container orchestration tool taking care of multiple boxes. For that purpose, we selected Docker Swarm among many tools including Kubernetes and Fleet, because Docker Swarm is very easy to install and use, also, we could alleviate our efforts on applying research developed with Docker into Docker Swarm. For that purpose, a version of a VIM and VNFM were developed to support Docker Swarm Clusters and expose them as a Point-of-Presence (PoP) in Open Baton. However, Open Baton's current version of the Docker Swarm VIM/VNFM does not support communication between two containers, part of the same deployment, in different clusters due to the lack of multi-cluster support in the default network driver of Docker Swarm. Especially, overlay networks configured by the default driver are isolated from the outside of a cluster.

One typical approach to enable containers in different sites to communicate with each other is to include multi-site boxes into a container cluster. Then the container orchestration tool could natively configure overlay networks across multi-sites. However, this approach has limitations for supporting use cases that require managing multiple clusters separately, for example, individual operation of each site or applying different policies to sites. Besides, adding multi-site worker boxes into a Docker Swarm cluster could degrade performance, resulting from manager-worker communication across multi-sites. Adding cluster managers, in contrast, results in performance reduction due to the consensus algorithm for synchronizing states among managers. Thus, this approach has limitations of scalability and performance in large-scale and widely distributed multi-site infrastructure. Therefore, OpenBaton's current version of Docker Swarm VIM/VNFM needs to be extended to support interconnection between containers running in multiple Swarm Clusters.

### 3 Requirements and Design

To explain our proposal, described in Sect. 1, we assume an example scenario that depends on multi-cluster service orchestration. This scenario consists of multiple Docker Swarm clusters registered to Open Baton as PoPs. Open Baton user selects one network service descriptor, which specifies the configuration and behavior of the virtual network functions of a service, through the Open Baton NFVO Dashboard. Then, the

user can select the clusters for each VNFs in the belonging to the network service. If the user selects the same cluster for all VNFs, then the service is a single-cluster service. If not, the service is a multi-cluster service. NFVO starts the deployment process according to the descriptor and the user's selection. In case of single-cluster, Open Baton creates an overlay network with the default networking driver of Docker Swarm. However, for multi-cluster service, Open Baton utilizes the Multi-Swarm Networking Helper to configure a Multi-Swarm Networking. After creating the network, Open Baton deploys the VNFs.

To realize the use case with the Multi-Swarm Networking Helper, Open Baton must consider the following:

- For a multi-cluster service, Open Baton utilizes Multi-Swarm Networking Helper to configure Multi-Swarm Networking.
- For a single-cluster service, Open Baton configures default Docker Swarm networking.
- Regardless of service types, all VNFs in a network service should work on an L2 overlay network.
- Do not modify common NFVO procedures for network service orchestration.
- Multi-Swarm Networking does not introduce any additional parameters in the Network Service Descriptor and VNF Descriptor.

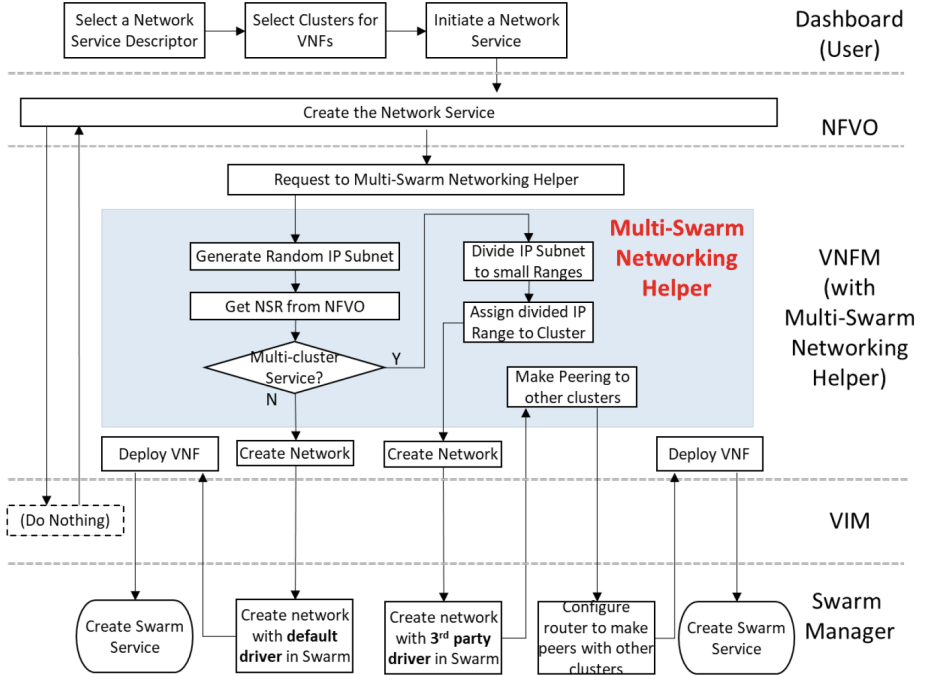
Before considering Open Baton, we had to find a way for Docker Swarm to enable an overlay network over multiple clusters. We considered three approaches: (1) to configure a relay container in each cluster. In this approach, all containers need to send packets destined to other clusters to the relay container working in the same cluster. The relay container can pass the packets to another relay container in the destination cluster; (2) to configure a Linux networking stack including Linux Bridge and internal firewall to inter-connect boxes in different clusters, whenever a container and a network is changed (creation, deletion, update). This approach requires the configuration of a forwarding table, neighbors table and a VXLAN tunnel on the Linux Bridge inside the network namespaces of the overlay networks; (3) to use the Weave Net driver that is one of the third-party network plugins for Docker Swarm. Weave Net driver manages an internal router in each box, and the router maintains networking information in the box such as the network list and attached containers. Those routers in the same cluster make peer relationships with each other and exchange the information. Then the driver configures an L2 overlay networking over the cluster according to the exchanged information. In this approach, we extend the scope of Weave Net router from a single cluster to multi-clusters by making peer relationships among Swarm manager boxes located in different clusters.

Among the candidates, we select to leverage on the Weave Net third-party driver. The third party driver could reduce the number of interaction between NFVO and Docker Swarm Boxes compared to other approaches. The driver required the addition of new steps to allow setting up the peers within Open Baton VIM/VNFM. After that, the driver automatically configures an overlay network over multi-clusters. In contrast, other approaches demand the NFVO to keep monitoring networking-related events in Swarm clusters, and directly configures Linux networking stack in all boxes whenever events occur. If the NFVO handles detailed configuration of all boxes, then advantages

by leveraging Docker Swarm as cluster resource orchestrator are decreased. For this reason, we decided to use the Weave Net driver approach, which is more suitable for our solution.

Implementing the approach into Open Baton has different issues. One of the main issues is duplicated IP addresses of containers in different clusters owing to default IPAM (IP Address Management) of Docker Swarm. Each Swarm cluster has a default IPAM. However, the IPAM does not know the IP addresses used in other clusters. Therefore, containers in different clusters may have the same IP address. To avoid this problem, Open Baton should divide L2 subnet into smaller IP allocation ranges and assign different range to each participating cluster to accommodate a multi-cluster service and reduce unassigned IP ranges, while creating an overlay network. Also, another design issue is to select the MANO component where we implement the feature for identifying multi-cluster services and calculating IP allocation ranges. The feature requires to find the number of clusters participating in the given service. We used the Docker Swarm VNF rather than the NFVO and the Docker Swarm VIM. The NFVO seems the most proper element since it can randomly generate an IP subnet for a network while knowing the number of clusters. Therefore, the NFVO can easily calculate IP allocation ranges and pass these ranges to the VIM/VNF when creating a network. However, it violates the requirement not to modify common NFVO procedures for the network service orchestration. By the way, the VIM does not receive any clues about the service being orchestrated by the NFVO, other than about the resources necessary to deploy a VNF. Thus, it cannot ask the number of clusters used by a network service to the NFVO. In contrast, VNF can find a service identifier from the given VNF, so it can query the NFVO to get information about the network service. In this context, we introduce Multi-Swarm Networking Helper doing all additional steps including service classification, network service descriptor query to NFVO calculation of IP allocation ranges, and creating multi-Swarm networks with the third-party driver. VNF can create both single-cluster and multi-cluster services with the help of Multi-Swarm Networking Helper.

The designed procedure using Open Baton for deploying a network service is shown in Fig. 1. Open Baton user selects a network service and the clusters for each VNFs, and then the NFVO starts the procedure for creating the network service. The NFVO requests the VIM to create an overlay network, the VIM, then skips this step, the reason is that the VIM does not know if the network belongs to a single-cluster network service or a multi-cluster one. For each VNF in the service, the NFVO sends a request with the VNF descriptor to the VNF to deploy it. The VNF then contacts the Multi-Swarm Networking Helper. The Helper extracts the service identifier from the given VNF descriptor and takes the network service record (NSR) that contains information of the service by sending a request to NFVO. The Helper classifies whether the service is deployed in single-cluster or multi-clusters, based on the cluster lists included in the received NSR. If the service is a single-cluster service, then the VNF creates a network using the default networking driver and the containerized VNF in the target cluster. If not, the Helper divides a subnet into multiple IP allocation ranges based on the number of clusters and returns one of the ranges to the VNF. VNF creates a network with the subnet using the assigned range with the Weave Net third-party driver. The Helper configures an internal router for making peers with the other



**Fig. 1.** Procedural design of deploying multi-cluster service with open baton MANO framework

clusters listed in the NSR. After those configurations, multi-Swarm networking becomes available, so VNFM creates the containerized VNF on the network. With this design, Open Baton satisfies the requirements previously established.

However, our design currently has limitations. We only consider the deployment procedure of service orchestration. We assume that multiple services do not share one overlay network, and all boxes in Docker Swarm clusters has pre-installed Weave Net third-party driver. Also, clusters configured with other software such as OpenStack cannot utilize it, because the design depends on Docker Swarm and Weave Net third-party driver.

## 4 Implementation and Verification

In this section, we describe the implementation of the Multi-Swarm Networking Helper in Open Baton based on the design proposed, and also verify its functionality on our testbed. For implementation and functional validation, we prepared a small-sized multi-site testbed consisted of two sites within the K-ONE (Korea OpenNetworking Everywhere) Playground. K-ONE Playground is a miniaturized multi-site Edge-Cloud testbed in South Korea. It consists of five sites, each of them comprising a K-Cluster that is a cluster consisted of multiple resource boxes. Those K-Clusters are interconnected through L3 WAN supporting 1 Gbps networking provided by KREONET research network [9].

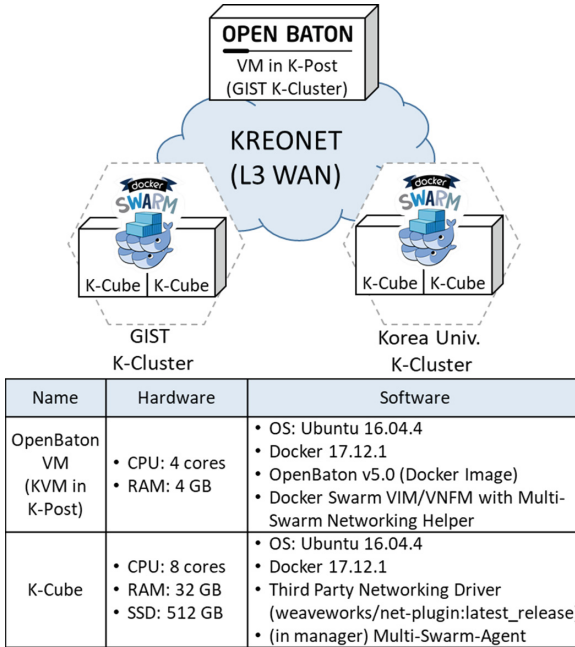
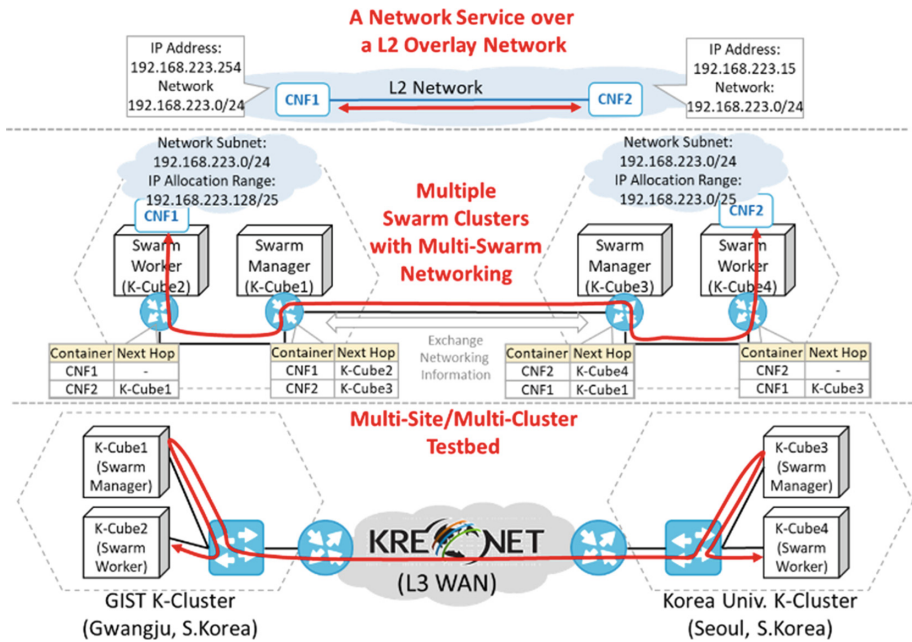


Fig. 2. The configuration of Multi-Swarm Networking testbed

Figure 2 shows the configuration of our testbed. Open Baton is deployed on an OpenStack VM (KVM) in K-Post box of GIST K-Cluster. We use a Docker image of Open Baton version 5.0 for deploying the NFVO. Docker VIM and VNFM work in the K-Post box, and they register to the NFVO via RabbitMQ. We used two K-Cube boxes from GIST (Gwangju, South Korea) K-Cluster and another two boxes from Korea University (Seoul, South Korea) K-Cluster. We configured the Docker Swarm clusters in two different sites and registered the clusters in Open Baton. Next, we installed Weave Net third-party driver to all boxes along with the Multi-Swarm-Agents in the Swarm Manager boxes. The Multi-Swarm-Agent acts as an intermediary between MANO and the third-party driver in the Swarm Manager box. The agent provides a REST APIs to Open Baton and configures the internal routers according to received requests. As a result, we ended up with Docker Swarm clusters in two different sites, being network services orchestrated upon them by Open Baton with Multi-Swarm Networking Helper.

In this testbed, we verify the functionality of Multi-Swarm Networking Helper by showing a simple scenario. We deploy a network service consisted of two containerized VNFs configured to be deployed in different clusters via Open Baton NFVO dashboard. For each CNFs, we used a customized Docker image of Ubuntu OS with networking test tools. Then, Open Baton NFV MANO with Multi-Swarm Networking Helper automatically configures an L2 overlay networking over two sites, so two CNFs can do L2-based networking each other.



**Fig. 3.** Verification of multi-cluster service orchestrated by Open Baton MANO with Multi-Swarm Networking Helper

Figure 3 shows the result of the functional validation. From a service perspective, two CNFs in different clusters have IP addresses of the same subnet and can directly communicate with each other through the L2 network. In what concerns Docker Swarm, two clusters had the networks created with Weave Net third-party driver. The networks, despite being on the same subnet, they are using different IP allocation ranges. For intra-cluster networking, internal routers in a cluster exchange networking information just after the network creation, and the third-party driver configures VXLAN-based overlay network over the boxes of the cluster accordingly. For Multi-Swarm networking, the Multi-Swarm Networking Helper makes peer relationships between routers in the Swarm manager boxes of the different clusters. The peered routers exchange information and spread it to Swarm worker boxes. All routers know the next hop router for packets destined to containers in different clusters. After the exchange, the driver creates VXLAN tunnels among routers in the Swarm manager boxes. As a result, an L2 overlay network extends to multiple Swarm clusters. Consequently, Open Baton NFV MANO can orchestrate multi-cluster services with Multi-Swarm Networking Helper and Weave Net third-party driver. Furthermore, CNFs of network services always work in the same way regardless of their location in underlay clusters.



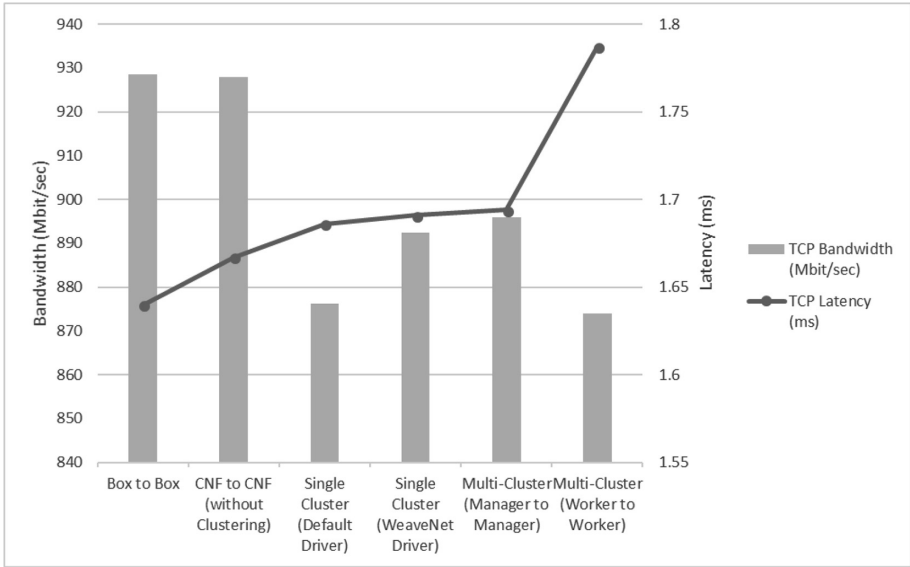


Fig. 4. TCP networking performance in different cases of multi-site container networking

We perform an additional experiment for measuring TCP bandwidth and latency in different cases of multi-site networking to verify the performance. To measure TCP performance, we use Qperf (version 0.4.11)<sup>1</sup> that is a benchmark tool designed for TCP/UDP as well as RDMA and other protocols. We consider six test cases of measuring TCP performance: (1) typical TCP/IP networking between bare metal boxes in different sites, as a reference point for other cases. (2) TCP/IP networking between two containers without clustering. (3) overlay networking with the default driver and single cluster over two sites. (4) overlay networking with Weave Net driver and single cluster over two sites. (5) multi-Swarm networking configured by multi-Swarm networking helper, and containers in manager boxes of two clusters in two sites. (6) multi-Swarm networking, and containers in worker boxes of two clusters in two sites.

Figure 4 shows the result of the performance measurement. Comparing the case 2 and 3 shows that overlay networking in Docker Swarm cluster decrease networking performance, due to network namespaces, virtual switches, internal firewall rules and virtual extensible LAN (VxLAN) tunnels additionally configured in each clustered box. Meanwhile, default driver and the third-party driver working under the same configuration have the equivalent performance as shown in the case 3 and 4. Multi-cluster service, in the case 5 and 6, shows TCP networking performance can be slightly changed according to locations of containers. As we explained in the feature verification, all packets destined to other clusters are sent to the manager box, and its internal router routes them to another manager box in the destination cluster. So, the additional hops are added to a networking path crossing the clusters. The results show that the

<sup>1</sup> <https://github.com/linux-rdma/qperf>.

additional two hops decrease approximately 20 Mbit/sec of TCP bandwidth and increase 0.1 ms of TCP latency. However, the amount of the decrease tends to be stationary, because the number of additional hops is at most two and the hops are between boxes in a cluster. Thus, we anticipate the overhead of multi-Swarm networking accounts for a relatively small portion of the performance degradation in large-scale infrastructures, where sites are widely distributed, and inter-site traffic is massive. Consequently, we insist on our solution, multi-Swarm networking orchestrated by OpenBaton MANO framework, has reasonable networking performance to support multi-cluster services.

## 5 Conclusion and Outlook

In this paper, we described a new approach to enable an overlay networking over multiple Swarm clusters with Weave Net third-party Docker networking driver to allow communication between containers. We also discussed the design and implementation of Multi-Swarm Networking Helper in Open Baton, automating the configuration of Multi-Swarm networking. We verified that Open Baton was able to deploy multi-cluster services with the support of our solution, and multi-Swarm networking has reasonable networking performance for multi-cluster services.

However, we only covered multi-cluster service deployment that is one part of the orchestration process. Therefore, we will improve Multi-Swarm Networking Helper to support all aspect of orchestration for Multi-Cluster services. Besides, we plan to implement a VIM and VNFM for Kubernetes that is the de-facto container orchestration engine in the market. After that, we will work on deploying CNFs over heterogeneous container-based clusters.

**Acknowledgements.** This work was supported by Institute for Information & Communications Technology Promotion (IITP) grants funded by the Korea government (MSIT) (No. 2015-0-00575, Global SDN/NFV Open-Source Software Core Module/Function Development and No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats), by Fraunhofer FOKUS and Universidade Estadual de Santa Cruz. Shin acknowledges Erasmus Mundus TEAM Programme for financial and administrative supports during mobility in Fraunhofer FOKUS.

## References

1. ETSI: Network Function Virtualisation (NFV); Terminology for Main Concepts in NFV. ETSI GS NFV 003 v1.2.1, ETSI Group Specification, December 2014
2. Carella, G.A., Magedanz, T.: Open Baton: a framework for virtual network function management and orchestration for emerging software-based 5G networks. IEEE Newsletter, July 2016
3. Felter, W., Ferreira, A., Rajamony, R., Rubio, J.: An update performance comparison of virtual machines and Linux containers. In: Proceedings of 2015 IEEE International Symposium on Performance Analysis of Systems and Software, pp. 171–172. IEEE (2015)

4. Cziva, R., Pezaros, D.P.: Container network functions: bringing NFV to the network edge. *IEEE Commun. Mag.* **55**(6), 24–31 (2017)
5. Francescon, A., Baggio, G., Fedrizzi, R.: X-MANO: cross-domain management and orchestration of network services. In: *Proceedings of 2017 IEEE Conference on Network Softwarization*. IEEE (2017)
6. Vilata, R., et al.: Fully automated peer service orchestration of cloud and network resources using ACTN and CSO. In: *Proceedings of Optical Fiber Communications Conference and Exhibition*. IEEE (2017)
7. Bonafiglia, R., Castellano, G., Cerrato, I., Risso, F.: End-to-end service orchestration across SDN and cloud computing domains. In: *Proceedings of 2017 IEEE Conference on Network Softwarization*. IEEE (2017)
8. ETSI: Network Function Virtualisation (NFV); Management and Orchestration, ETSI GS NFV 001 v1.1.1. ETSI Group Specification, December 2014
9. Shin, J., Kim, J.: K-Cluster-based Playground for SDN/NFV/Cloud Integration. In: *Proceedings of the 27th Joint Conference on Communications and Information, KICS* (2017)