Demonstration of Containerized vDU/vCU Migration in WDM Metro Optical Networks

Jiaxin Feng[1*], Jiawei Zhang[1], Yuming Xiao[1], and Yuefeng Ji[1]

[1]*State Key Lab of Information Photonics and Optical Communications, Beijing Univ. of Posts and Telecommunications (BUPT), Beijing, China; Email: {fengjiaxin*, zjw, yumingxia, jyf}@bupt.edu.cn*

Abstract:
We experiment on a containerized vDU/vCU migration for load balancing among processing pools over WDM metro networks. Two stateful migration strategies to reduce migration time are verified on a converged edge access network platform.
OCIS codes: (060.4250) networks; (060.4254) Networks, combinatorial network design.

1. Introduction

With the continuous growth of user terminals, an important challenge in future radio access network (RAN) is to sustain an exceptionally higher traffic volume using limited resources. Cloudification and virtualization of RAN are becoming a consensus for RAN deployment, which can promote the flexible resource reuse [1]. Baseband processing functions, such as Distributed Unit/Central Unit (DU/CU), can be virtualized and deployed as virtual network functions (VNF) on a general-purpose platform (GPP) in processing pools (P-Pool) [2,3]. To improve the resource utilization, virtualized DU/CU (in Virtual Machine or Container) should be placed with fewest activated P-Pools [4]. However, in real operation networks, mobile traffic presents a strong time-geometry fluctuation where underloaded and overloaded P-Pools coexist in networks. Therefore, it is essential to migrate these vDU/vCU between P-Pools for more service acceptance. Both resource utilization and user experience can be improved. To achieve this, two challenges have emerged: 1) the forwarding plane is programmable; 2) RAN network functions (NF) can be flexibly deployed and migrated.

Some solutions are recently proposed with the introduction of SDN/NFV into telecommunication networks. Authors in [5] demonstrated flexible interconnection of RAN NFs, such as virtual DU (vDU) and CU (vCU) in a RAN-as-a-Service (RANaaS) scenario. Authors in [6] provided a joint SDN/NFV orchestration to enable the deployment of the network function. However, the existing researches are mainly designed to control forward plane for VNF deployment, whereas flexible VNFs migration and management haven't been considered.

In this paper, we introduce VNF migration mechanism into NG-RAN, and provides a joint orchestration framework over WDM network for the forwarding control and VNF management. In Converged Edge Access Network Platform (CEANP) testbed, we experimentally demonstrate the migration of containerized DU/CU between P-Pools to achieve the load balancing.

2. Description of vDU/vCU Migration and System Frameworks

Fig. 1(a) shows the demonstrated NG-RAN architecture with WDM connections from remote unit (RU) to containerized vDU and vCU. Container-based DU/CU deployment is an attractive alternative: on the one hand, they provide a similar level of isolation compared to Virtual Machines (VMs), on the other hand, containers are able to offer superior I/O and CPU performance than Kernel-based Virtual Machine or VirtualBox [7].

In this demonstration, distributed P-Pools responsible for providing memory and CPU resources for vDU and vCU. And it must respond to factors such as user mobility and fluctuating load to maintain energy efficiency and quality of service. Existing works consider benefits in terms of cost and power consumption to deploy DUs and CUs into few P-Pools [4]. As the user aggregates, P-Pool2 is over-loaded. To maintain quality of service, vCU2 is migrated to P-Pool3 for balancing the P-Pool resources. To achieve stateful migration of vDU/vCU, we define the process of migration as two steps: ***Step1***: vDU/vCU migration: (1) Collect vCU2 container running status in P-Pool2(such as process tree, memory page, namespace, cgroups, shared memory, network, etc.) and freeze it as checkpoint files in disk ; (2) Transfer the checkpoint file data to the target P-Pool3; (3) In target P-Pool3, restoration the vCU3 through rebuilds the process tree and memory by using checkpoint data and container images. ***Step2:*** Lightpath reconfiguration: In order to maintain the full state of the RAN, the light path needs to be reconfigured to ensure that the RU2 can reach the location of vDU2 and vCU2.

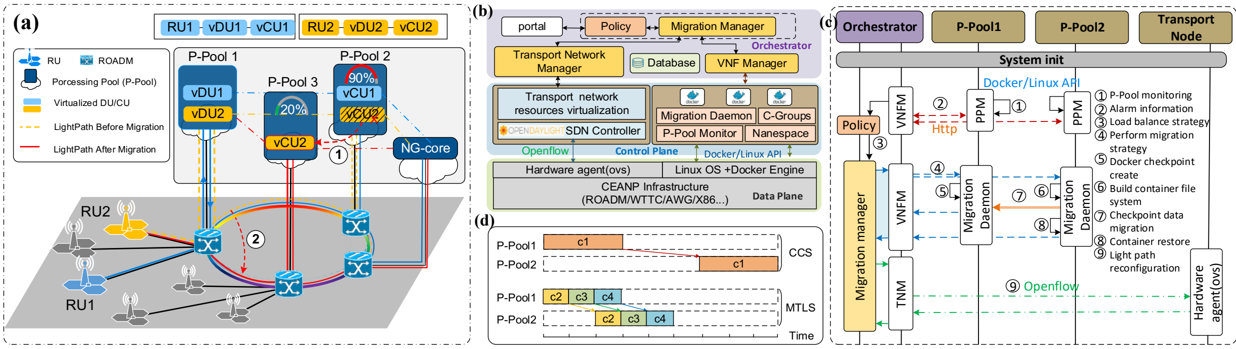

Fig.1(a) Illustration of vDU/vCU migration;(b) orchestration and control framework;(c)cooperation procedures for DU/CU migration

The proposed framework depicted in Fig. 1 (b). It consists of global orchestrator, a control plane, and data plane. In the data plane, X86 based hosts was virtualized by lightweight virtualization techniques to provide a running environment for the docker container. OpenFlow Agent (OFA) is connected to each real physical transport node and communicates with SDN Controller. In the control plane, P-Pool Monitor (PPM) and Migration Daemon (MD) are deployed in each P-Pool. The CPU and Memory of each P-Pool including each container is monitored by PPM. And MD is used to perform checkpoint creation, data transfer, file system construction, and checkpoint restore of the vCU that need to migrated. Orchestrator have a global perspective of networks and is strategy-programmable through modularity bundle based on OSGi dynamic module system. The Transport resource manager (TRM) and VNF manager (VNFM) components are responsible for collecting all the node information in the network, constructing the physical network topology stored in the database, and configuring the corresponding physical nodes. Migration Manager (MM) is responsible for the overall migration process coordination and orchestration. The policy module (POLICY) maintain the migration strategy which is programmable. Fig. 1(c) shows the detailed interaction between the modules of the framework during the vDU/vCU migration. The VNFM maintains a set of load thresholds and accept the P-Pool load condition information from PPM. When an overload or underload condition occurs, the alarm information will be sent by the VNFM to POLICY (Corresponding to steps 1-2). And then, migration strategy in POLICY will be executed through MM (Corresponding to steps 3-9). In detail, the container migration process is performed by VNFM (Corresponding to steps 3-8) and the light path reconfiguration process is performed by the TRM (step 9).

We use balance time to represent the total time of releasing a certain amount of resources. It is the sum of time includes multiple containers migration and optical path reconfiguration. The optical path reconfiguration time is in the millisecond level and the time bottleneck is in the migration. We studied the time loss in migration and expressed it as $T_{mig} = T_{ckp} + T_{trans} + T_{rest}$, where $T_{ckp} = a \cdot C_{i_r}^{mem}$ and $T_{rest} = c \cdot C_{i_r}^{mem}$ are the checkpoint creation and restore time, respectively , which is approximately equals to the memory size multiplied by the reciprocal of the disk copy rate. $T_{trans}^{i,j,c_i} = \frac{1}{B} \cdot c_{i_r}^{ckp} + T_t^{i,j}$ is the transmission time of the checkpoint data, which is approximately equal to the checkpoint size multiplied by the reciprocal of the network transmission rate plus the time of signaling interaction. Based on the above, we propose a “Minimize Transmission Blocking Time (MTBL)” scheme, and compare it with widely used strategies “Capacity Closest Strategy (CCS)”. Fig1. (d) illustrates the two strategies. The CSS algorithm obtains some containers through greedy strategies, and the sum of their resource occupancy is closest to the given release threshold. For MTBL, it selects some containers through a dynamic programming strategy, which minimizes the connection blocking time, thus minimize the balance time.

3. Experimental Setup and Results

Fi.3(b) shows the proposed demonstration topology. And we demonstrate the DU/CU migration on the CENAP testbed shown in Fig. 3(a). We use Wavelength Tunable Transponder Card(WTTC), switch and 1×9 wavelength selected switches (WSS) as a transmission devices. In addition, the open source project Opendaylight (ODL) acts as an SDN controller and we extended OpenFlow protocol for controlling and monitoring transmission the above equipment. The two processing pools are 64bit virtual machines with Linux kernel version 4.15.0 in two different servers, with 8G RAM and 4 CPU, and are virtualized by docker. DU/CU functions have been implemented in containerized diseases (such as open source Open Air Interface, OAI) [8] and is wildly utilized as mobile network platform [9].In order to verify the proposed migration strategy, we use multiple stress containers in the processing pool to consume CPU and Memory resources in the processing pool until the migration strategy is triggered.

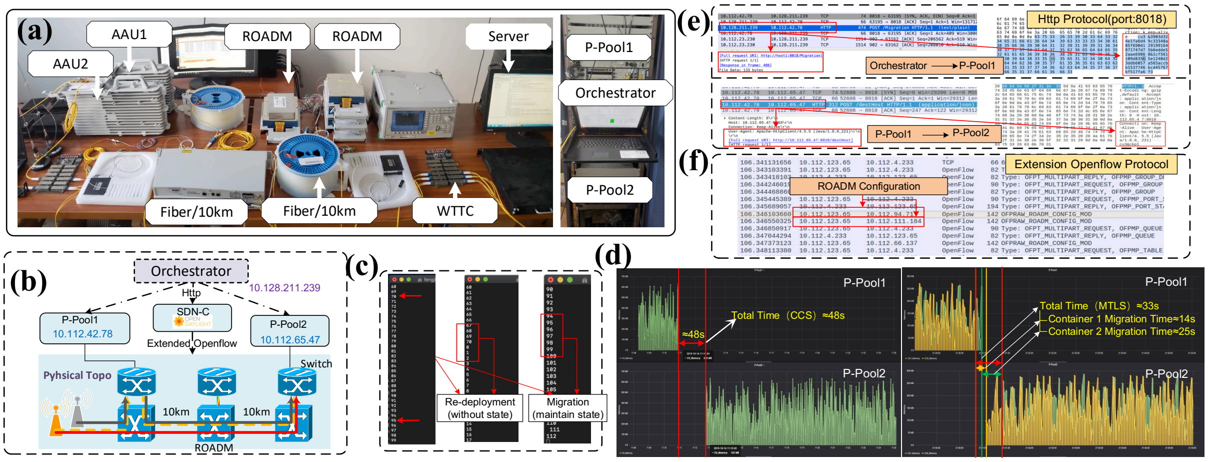

Fig. 2(a) CEANP testbed; (b) experimental topology;(c) Stateful Migration maintains process status;(d) Comparison of two migration strategies (e) Wireshark captures of interaction between orchestrator and Migration Daemon; (f) Wireshark captures of extended OpenFlow messages for ROADM reconfiguration;

As is shown in Fig.3 (c), we introducing process Checkpoint/Restore In Userspace (CRIU) tool for freezing running container. Compare to the re-deployment method, the restored container use stateful migration in P-Pool2 maintains its state when it was frozen. Fig. 3(d) shows the memory monitoring of the containers in two processing pools, and compared two migration strategies. In the CSS strategy, the container which capacity closest to the release capacity(512MB) is selected and migrated. The container migration time is 48s, which has a time advantage over traditional minute-level wireless reconfiguration time. The MTLS policy has a shorter migration time than CCS. It chooses two small memory-occupied containers for migration and the total balance time is 33s. The Fig. 3(e) shows the wireshark capture of the interaction between orchestrator and Migration Daemon in two P-Pools. The Migration Daemon listens to port 8018 to receive commands, which include information about container ID and target P-Pool info, sent by the orchestrator. After the container is frozen, the checkpoint file is sent between the processing pools via the SCP protocol. Fig. 3(f) shows the wireshark captures of the extended OpenFlow protocol messages from the controller to the hardware. The message indicates the ligthpath reconfiguration on the ROADMs separately, so that the signal of the RU is reconnected from the P-Pool1 to the P-Pool2. The ligthpath reconfiguration time is in milliseconds.

4. Conclusion

In this paper, we introduced the DU/CU Migration mechanism to NG-RAN to accommodate future RAN flexibility requirements. The migration mechanism based on the two strategies is demonstrated in CEANP to achieve load balancing. Compared to the traditional wireless reconfiguration strategy, the migration mechanism can maintain the status of the DU/CU while having a shorter service downtime.

Acknowledgment This work was supported by the National Key R&D Program of China (No. 2018YFB1800802), the National Nature Science Foundation of China Projects (No. 61971055, 61871051), the Beijing Natural Science Foundation (No. 4192039), the fund of State Key Laboratory of Information Photonics and Optical Communications, China, IPOC2019ZT05.

5. References

[1] Parvez, Imtiaz, et al. "A survey on low latency towards 5G: RAN, core network and caching solutions." IEEE Communications Surveys & Tutorials 20.4 (2018): 3098-3130.
[2] Kondepu, K., et al. "Orchestrating lightpath adaptation and flexible functional split to recover virtualized RAN connectivity." *Proc.* OFC. IEEE, 2018.
[3] Alessio Giorgetti, et.al “Demonstration of Fault Localisation and Recovery of Optical Connectivity Supporting 5G vRAN”, *Proc.*ECOC. IEEE, 2019.
[4] Xiao,et.al "Energy efficient Placement of Baseband Functions and Mobile Edge Computing in 5G Networks." Proc. ACP. IEEE, 2018.
[5] Sgambelluri, A., et al. "Provisioning RAN as a Service (RANaaS) Connectivity in an Optical Metro Network Through NETCONF and YANG." *Proc.*ECOC. IEEE, 2018.
[6] Bravalheri, Anderson, et al. "VNF Chaining across Multi-PoPs in OSM using Transport API." *Proc.* OFC. IEEE, 2019.
[7] Giannone, et al. "Impact of Virtualization Technologies on Virtualized RAN Midhaul Latency Budget: A Quantitative Experimental Evaluation." IEEE Communications Letters 23.4 (2019): 604-607.
[8] “OpenAirInterface”, gitlab.eurecom.fr/oai/openairinterface5g/wikis/home
[9] CRIU, [online] Available: https://criu.org/Main_Page.