Abdullaziz, O.I., Wang, L., Chundrigar, S.D., Huang,K. (2018). ARNAB: Transparent Service Continuity across Orchestrated Edge Networks.

ARNAB: Transparent Service Continuity across Orchestrated Edge Networks

Osamah Ibrahiem Abdullaziz*, Li-Chun Wang†, Shahzoob Bilal Chundrigar‡ and Kuei-Li Huang§
*†Department of Electrical Engineering & Computer Science, National Chiao Tung University, Taiwan
*yabolahan.04g@g2.nctu.edu.tw, †lichun@g2.nctu.edu.tw
‡§Information and Communications Research Laboratories, Industrial Technology Research Institute, Taiwan
‡shahzoob@itri.org.tw, §garyhuang@itri.org.tw

***Abstract*—In this paper, we present an architecture for transparent service continuity for cloud-enabled WiFi networks called ARNAB: ARchitecture for traNsparent service continuity viA douBle-tier migration. The term arnab means *rabbit* in Arabic. It is dubbed for the proposed service architecture because a mobile-user service with ARNAB behaves like a rabbit hopping through the WiFi infrastructure. To deliver continuous services, deploying edge clouds is not sufficient. Users may travel far from the initial serving edge and also perform multiple WiFi handoffs during mobility. To solve this, ARNAB employs a double-tier migration scheme. One migration tier is for user connectivity, and the other one is for edge applications. Our experimental results show that ARNAB can not only enable continuous service delivery but also outperform the existing work in the area of container live migration across edge clouds.**

***Index Terms*—Container live migration, Virtual access point, WiFi handoff, Service continuity, MEC, SDN, NFV.**

I. Introduction

Wireless networks will soon be the key enabler for ultra-reliable and low latency services. Wireless communication systems are driven by the quality of services they provide to the end users. Although the industry verticals such as automotive and mobility, robotics and automation, e-health, and entertainment applications create new business opportunities for the network operators, they pose challenges in terms of deployment cost, reliability and latency requirements. For instance, augmented reality application will revolutionize the entertainment industry because of the realism it brings to the user experience. The delivery of good experience and the feel of realism requires small round trip time (RTT) for action and reaction [1].

To meet the challenging requirements of these applications, multi-access edge computing (MEC) and network function virtualisation (NFV) have emerged as solutions where the cloud services are brought closer to the edge of the network. On the one hand, MEC reduces backhauling and end-to-end RTT. On the other hand, NFV decouples the network functions and applications from the underlying hardware and allows them to be implemented as software.

Now that services can be deployed independently from the hardware in a distributed fashion, software define networking (SDN) enables a dynamic and responsive network to new services. SDN complements MEC and NFV especially due to the separation of the control and data planes which simplifies management, provides programmability and enhances scalability and performance. Together, SDN, MEC and NFV technologies empower operators to efficiently orchestrate and scale their networks.

The integration of these technologies is demonstrated by 5G-CORAL [2] which leverages edge and fog computing to create unique opportunity for access convergence. It contemplates ETSI NFV [3] and ETSI MEC [4] standards while also taking into account mobile and volatile computing, networking and storage resources in a multi-RAT environment. 5G-CORAL architecture consists of two major building blocks as shown in Fig. 1. The edge and fog computing system (EFS) contains edge and fog resources that belong to a single administrative domain. An EFS provides service platforms, functions, and applications on top of virtual resources, and may interact with other EFS domains. The orchestration and control system (OCS) is responsible for managing and controlling the EFS.

In WiFi networks, SDN and NFV are exploited to simplify management through programmable wireless devices and reduce the cost of deployment through network functions virtualization. Also, WiFi infrastructure could benefit from MEC paradigm to offer high quality real-time applications. For example, cloud services can run inside lightweight containers, such as Linux containers (LXC), on or near wireless access points.

To maintain the benefits brought by MEC in a cloud-enabled WiFi network, especially during user mobility, we identify the following challenges.

- In the standard 802.11 protocol, handoff process takes up to 2s [5] during which the ongoing communication sessions may be dropped. This becomes an issue for highly interactive applications.
- Containerization technology improves resource utilization but container live migration is not mature to support MEC environment.

In this paper, we propose ARNAB architecture to address these challenges. ARNAB provides service continuity through double-tier migration, namely user connectivity migration and edge application migration. In principle, the first tier migrates client connectivity to avoid handoff interruption while the

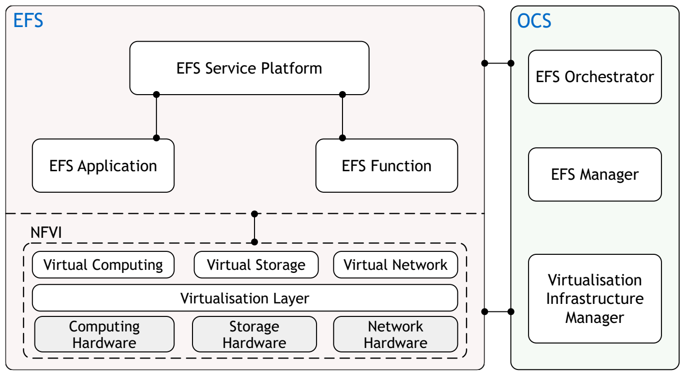

Figure 1: 5G-CORAL reference architecture.

second tier migrates containerized applications between edge clouds.

The contributions of this work are summarized as follows.

- Proposed ARNAB architecture which provides transparent service continuity for users in cloud-enabled WiFi infrastructure. To the best of our knowledge, ARNAB is the first work that aims at orchestrating connectivity and application migration simultaneously.
- Developed a pre-copy procedure to reduce service downtime when migrating containerized applications in MEC environment. This work outperforms the existing solution presented in [6].
- Evaluated ARNAB's connectivity and application migration on a real-world environment.

The rest of the paper is organized as follows. Section II reviews the existing work in connectivity and application migration. Section III presents the ARNAB architecture. Section IV provides and discusses the experimental results. Finally, we state our concluding remarks and future direction in Section IX.

II. Related Work

Here, we review the existing work and point out the research gap to achieving a continuous service delivery across edge clouds.

A. *WiFi Handoff and Connectivity Migration*

In the standard IEEE 802.11, the association procedure begins with a discovery phase, where clients actively scan for available APs. During the scan, APs that respond to probe messages become candidates for association. Once an AP is selected, the association is defined between the client's MAC address and the basic service set identifier (BSSID) of the AP. In this procedure, the infrastructure has no control over the client association decision. Therefore, to change an AP, the client starts the handoff process which may take up to 2s [5]. To reduce the re-association time, many schemes for fast handoff are proposed in [7]–[11]. These schemes can be classified into scan-time reduction and authentication-time reduction. In the scan-time reduction, the aim is to identify a target AP for handoff as fast as possible. Examples for scan-time reduction schemes include sync scan [7], intelligent channel scan [8], neighbour graph [9], selective neighbour caching [10], AP prediction [11] and IEEE 802.11k. In authentication-time reduction, the concept of pre-authentication is introduced [11] and also specified in the IEEE 802.11r. Although the proposed schemes can reduce the re-association delay, those schemes require modification to the client side and introduce additional signaling. This fact challenges the concept of bring-your-own-device, which implies that the infrastructure should accommodate diverse range of user devices.

To move the client-AP association decision from the client to the infrastructure, virtual access point (vAP) is introduced in [12]. The vAP is a network function abstraction which is created for connecting clients. Each client will be associated with a dedicated vAP, which consists of the following parameters: (1) client MAC address, (2) client IP address, (3) a fake BSSID, and (4) service set identifier (SSID), to be used in the communication.

Based on [12], a client associates with a vAP and periodically receives a beacon to keep it aware that its AP is still available within range. The signal-level received from the client is stuffed in the beacons so that the neighbouring APs can over-hear it. Every AP maintains two lists that stores client signal-level, managed and monitored lists. The managed list contains the clients that are currently associated with the AP and the monitored list contains clients that the AP can hear. Connectivity migration occurs when a neighbouring AP receives a beacon from the client with a signal-level higher than what is advertised by the serving AP. This approach moves the association decision to the infrastructure but has drawbacks. All the APs are assumed to operate on the same channel to be able to hear the advertised signal-level. This makes the solution impractical for real deployment and frequency planning. In addition, there is a lack of global view since the management of vAPs is done in a distributed manner.

Later, a multichannel vAP handoff extension of [12] is proposed in [13]. In this solution, APs operate in different channels and communicate with each other to manage client mobility. , After a client connects to a vAP which is managed by an AP, the AP monitors the client signal-level and sends a scan request to the neighbouring APs if the signal-level goes below a threshold. When a neighbouring AP responds to the request, the client is forced to switch channel and continues the communication with the new AP. This scheme overcomes the interference issue caused by operating in the same channel but still lacks a global view of the entire infrastructure.

Alternatively, a software defined WiFi network framework called Odin is proposed in [14]. Odin integrates SDN-based solutions to the concept of vAP. That is, programmability and global view of the network are available for the management of client mobility. In [15], Odin framework is utilized to move vAP between different APs while client is generating game traffic. Although the schemes in [14], [15] offer flexibility and scalability in management, they still assume that all the APs operate in the same channel. More recently, a solution that combines both the advantage of operating in multichannel environment and global view based on SDN is presented in [16], [17]. In ARNAB, we follow the same approach presented in [16] to enable continuous delivery of services at network access.

B. Container Live Migration

Service migration can be classified into stateful and stateless. In stateless migration, the state of the service is not preserved when the service is moved to the destination host. In the case of stateful migration (also known as live migration), the state of the service is retained when the execution of the service is resumed at the destination host. There are three types of stateful migration techniques, namely stop-then-copy [18], pre-copy [19] and post-copy [20]. First, stop-then-copy freezes a service, dumps its state, copies the service and its state to the destination then restores the service. Second, pre-copy performs iterative state dumping while the service is running then concludes with a shorter stop-then-copy. Third, post-copy performs a short stop-then-copy to move essential data, then starts the service at the destination and retrieves the rest of the data when needed.

VM live migration is well investigated [21] and many effective solutions are commercially available. For instance, a pre-copy based VM live migration scheme is presented in [19]. An active VM continues to run in the course of in-memory data iterative pre-copying. During a consecutive iteration, only changes in memory (dirty pages) are transferred. At last, a final state copy is performed while the VM instance is frozen and then transferred to the destination host. This way, the amount of downtime is greatly reduced when compared to a pure stop-and-copy scheme. Although the work in VM migration is mature, most of the existing solutions are tailored for data centre environment where network-attached storage (NAS) and specific virtualization technology are utilized. NAS enables all the host machines in a data centre to access a network-shared storage which removes the need for migrating disk storage. However, in a scenario where migration takes place between MECs, state and local-disk storage has to also migrate over wide area network (WAN).

Lately, container migration has caught much attention from the research community [22] [6]. Especially, since containerization offers many advantages, in terms of resource efficiency and performance, over traditional hypervisor-based virtualization. This fact enables the instantiation of lightweight containerized applications suitable for IoT services [23]. In [22], container migration mechanism is developed for power efficiency optimization in heterogeneous data centre. This work assumes that the source and destination hosts have access to a NAS and thus container data is not copied over WAN.

Furthermore, a framework for migrating containerized applications is presented in [6]. The proposed framework is the first to consider MEC environment for container migration. Fundamentally, the framework is a layered model which aims to reduce the downtime incurred by the migration process. While the presented results shows reduction in downtime as a result of layering, the framework relies on stop-and-copy migration which is not an efficient method for containers with large in-memory state. In our proposed solution, we develop a pre-copy procedure to migrate containerized applications between edge clouds.

III. The ARNAB Architecture

In this article, we propose ARNAB, which stands for: Architecture for transparent service continuity via double-tier migration. The term ARNAB means *rabbit* in Arabic and it is dubbed for the architecture because it illustrates the behaviour of the service hopping through the infrastructure following the user. Moreover, ARNAB is transparent to the user since it does not require any modification to the user device. The ultimate goal of ARNAB is to provide a seamless user experience through continuous service delivery. ARNAB employs double-tier migration, namely user connectivity migration and edge application migration. On the one hand, the first tier utilizes vAP to avoid WiFi handoff interruption and moves the association decision from the clients to the infrastructure. On the other hand, the second tier utilizes pre-copy procedure to reduce the downtime of containerized application migration. Fig. 2 illustrates the architectural model of ARNAB.

A. Tier 1: User Connectivity Migration

ARNAB leverages innovative technologies, such as SDN, Odin framework [15], click modular router [24] and OpenWRT [25], to deliver seamless WiFi experience using commodity hardware. Odin for ARNAB consists of Odin master and Odin agent, and provides two southbound protocols namely OpenFlow and Odin protocol. The Odin master, which is part of the OCS, utilizes OpenFlow protocol to install flow rules in APs to steer traffic when vAPs are migrated. In addition, the Odin master utilizes Odin protocol to communicate with Odin agents. The commodity WiFi APs, used in the experimental setup, have two wireless interfaces and run click modular router on OpenWRT embedded operating system with Odin agent elements. Finally, Odin mobility application, which runs on top of the Odin master, manages the connection and the mobility of clients including vAPs assignment during association, and vAPs migration during mobility.

User connectivity migration logical steps and example are detailed as follows.

Step 1: *vAP-association* - a client $C2$ scanning for a network will be associated with a dedicated vAP_{C2}, which is spawned within the Odin agent running on AP_1 operating in channel CH_6 at the proximity of the client.

Step 2: *vAP-keep* - vAP_{C2} keeps the client attached by periodically unicasting beacon frames to the corresponding client to keep it informed that AP_1 is still at its radio reach.

Step 3: *RSSI-monitoring* - AP_1 keeps track of the received signal strength indicator (RSSI) from $C2$ and informs the Odin master when the RSSI is below a threshold.

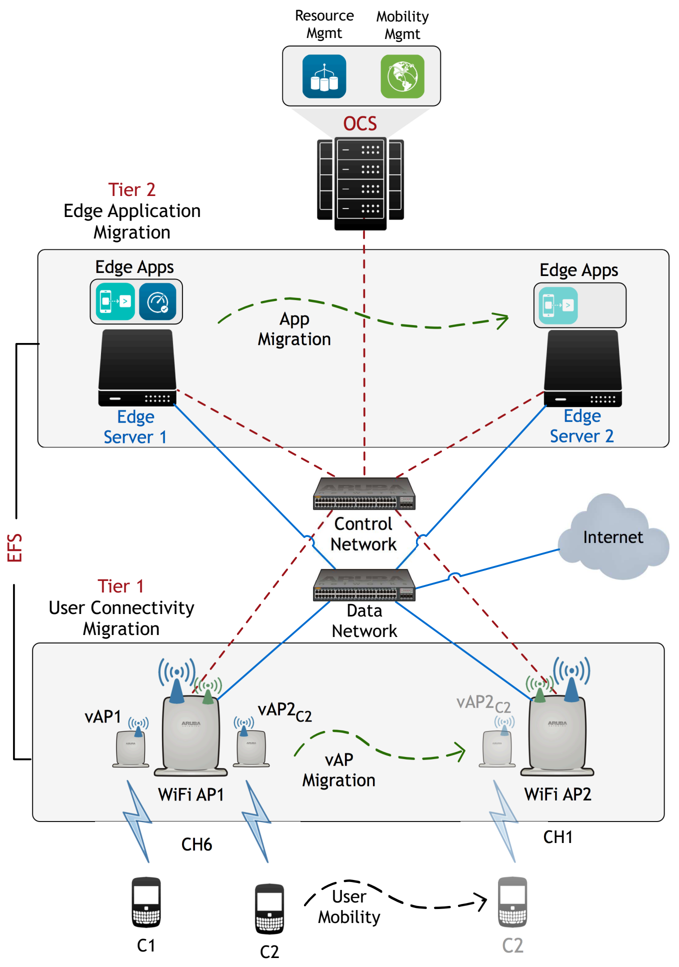

Figure 2: Architectural model of ARNAB.

Step 4: *vAP-migration* - Odin master instructs the APs near the client to switch their secondary interface to CH_i to listen to the client's packets. Then, the Odin master identifies the best candidate AP for vAP migration in this example AP_2 operating in CH_1 is selected. Next, the Odin master instructs AP_1 to tell $C2$ to switch to channel CH_1. Concurrently, the Odin master initiates vAP_{C2} in AP_2 which begins sending beacons to $C2$. Finally, once $C2$ has switched to CH_1, its connection is handed over to AP_2 and the vAP_{C2} in AP_1 is removed.

From the client's perspective, the AP which it is associated with has not changed. In reality, its vAP is still the same but its connectivity has migrated to a different physical AP. In such way, the handoff decision is moved from the client device to the network infrastructure. At the same time, the handoff is no longer required to support user mobility. Instead, channel switch and its relatively small delay is what the client experiences when changing APs.

B. Tier 2: Edge Application Migration

ARNAB application migration enabling technologies include 5G-CORAL, Linux container (LXC), checkpoint and restore in userspace (CRIU), and remote synchronization (rsync). The OCS manages the lifecycle of the containerized application in the EFS such as instantiation, cloning, migration, scaling and termination. CRIU is utilized to dump the state of the migrating containers. The local-disk and the state of the containers are transferred by utilizing rsync for its remarkable speed and efficiency.

To migrate a container between edge nodes with minimal downtime, ARNAB develops a pre-copy procedure, which is summarized in following steps:

Step 1: *Local-disk-copy* - the container base-image is assumed to be available in all edge nodes to reduce traffic overhead and to keep the total migration time to minimal. Local-disk synchronization is performed to copy application related files.

Step 2: *Iterative-pre-copy* - all the pages of the container including the running applications are dumped then transferred to the destination while the container continues to run. Next, a number of pre-copy iterations dumps and transfers only the memory pages that have been changed (dirtied) since the last dump.

Step 3: *Stop-then-copy* - the container gets frozen in this step then a final dump and copy is performed. The downtime observed by the user occurs during this step.

Step 4: *Restore-and-terminate* - the container is restored in the destination edge node and the frozen container in the source edge node gets terminated.

It is important to note that an edge node and an AP can be the same physical device that may possess communication, computing and storage capabilities. Also, the migration of vAP is not a trigger for the migration of containerized applications. Currently, the migration of vAP depends on the signal strength received from the connected clients. Whereas, the migration of containers depends on the applications latency and bandwidth requirements. The optimization of the application migration decision is not the focus of this article. Decision optimization solution such as [26] can be adopted.

IV. Experimental Results and Discussion

The experimental setup of ARNAB is illustrated in Fig. 2 and consists of a server running OCS, two edge servers where the containerized applications run, a client, two WiFi APs with two wireless interfaces and two Ethernet switches, one for control traffic and one for data traffic. The hardware and the software specifications of the setup are detailed in Table I. This experiment has two targets namely, 1) measure the impact of the user connectivity migration on application throughput, and 2) measure the observed downtime during edge application migration.

To measure the impact on application throughput due to connectivity migration, a video streaming server is installed in a container. This server is made accessible through the client browser. Fig. 3 shows the impact of the vAP migration on the application throughput. The dashed segment of the

Table I: Hardware and software specifications used in the experimental setup.

Edge Servers	Hardware	Model	ProLiant DL160
		CPU	Intel Xeon 2.10GHz
		RAM	124GiB DIMM 2400
		Network	2 × I350 Gigabit
	Software	OS	Ubuntu 16.04
		Kernel	4.4.0-119-generic
		LXC1	3.0.0
		CRIU	3.8
Access Points	Hardware	Model	TL-WR1043ND v2
		Platform	Qualcomm Atheros
		Flash	8MB + (16GB)
		Network	4×Eth 2×Wireless
	Software	OS	OpenWRT (15.05.1)
		Kernel	3.18.23
		OVS	2.8.2
		Click	2.1
Container	Image	Arch.	amd64
		OS	Ubuntu 16.04
		Release	Xenial
		Kernel	4.4.0-116-generic

plot represents the application throughput while the client is associated to vAP_{C2} running in AP_1 operating in CH_6. The dotted segment of the plot represents the application throughput after the migration when the client is associated to vAP_{C2} running in AP_2 operating in CH_1. Because the signal-level reached the specified threshold, the OCS migrates vAP_{C2} from AP_1 to AP_2. Note that, the client sees a channel switch so the client TCP session with the streaming server continues after the migration. The impact on the throughput is very short which is an expected result due to channel switching. Channel switching delay of the same APs used in our experiment is reported in [16] to be approximately between $90ms$ to $110ms$ at different inter-beacon time.

To benchmark container migration, we implemented stop-then-copy migration to reproduce the results presented in [6] and evaluated its downtime against our pre-copy migration scheme. The migration experiments were carried out between the edge servers. Table I also includes the specification of the container used in this experiment. In this study, we evaluate the downtime during the migration of two containerized applications, video streaming and ram simulation applications. Ram simulation represents those applications with intensive use of system memory (i.e., high rate of page dirtying). Examples of ram intensive application include in-memory database, big data analytics and deep learning. To simulate such application, a very small program is used to load the container memory with approximately 1GB of data then manipulates the loaded data to reflect page dirtying. This is particularly important when pre-copy migration is utilized. The container state size of every iteration in pre-copy procedure depends on the rate of dirty pages. Fig. 4 shows the service downtime of LXC containers when using pre-copy and stop-then-copy migration. The results are based on average values of 10 trails for each presented case. In the case of video streaming application, pre-copy shows minimal improvement over stop-then-copy. Most

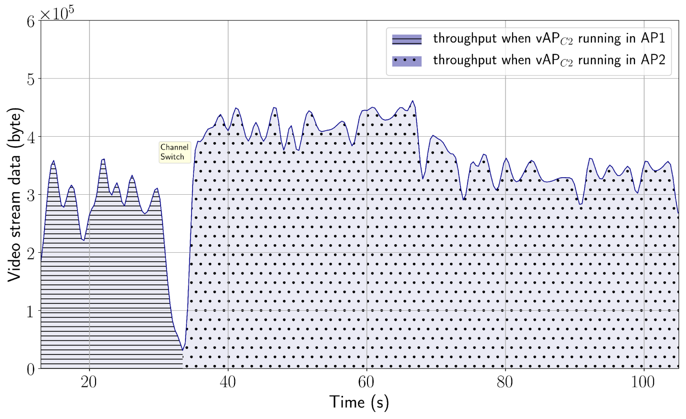

Figure 3: Throughput impact of video streaming application during vAP migration.

of the downtime is spent in the common steps (i.e., final dump $\rightarrow$ state transfer $\rightarrow$ restore) of both migration schemes rather than the amount of data copied. In the case of ram simulation, note that the downtime of stop-then-copy is much higher than pre-copy due to the amount of data being transferred while the container is frozen. For example, when 20% of the in-memory data frequently changes, the downtime of stop-then-copy and pre-copy are $14.7s$ and $5.4s$ during state copy of 1.2GB and 171MB, respectively. Furthermore, as the rate of dirty pages increases, the downtime of pre-copy slightly increases but still outperforms the stop-and-copy.

V. ARNAB USE CASES

In this section, we present few use cases that could leverage ARNAB architecture to improve user experience and fulfil real-time application requirements. Here, the presented use cases are not limited to cloud-enabled WiFi networks. Cloud-enabled cellular networks can also benefit from the support of live container migration across edge clouds.

- **Augmented Navigation**: Service continuity is essential for augmented reality based indoor navigation applications. Augmented navigation applications for public places, such as shopping malls, museums, airports and exhibition centres, can utilize ARNAB's double-tier migration to support user mobility. That is, the infrastructure hosts containerized application for each user and maintains the state and connectivity while the user roams through the network.
- **Cloud Robotics**: There are various cases where cloud robotics can be used to handle tasks in indoor environments such as factories, warehouses and also shopping malls. For example, in shopping malls, robots can be used for security, operational assistance, goods transport. In addition, robots can be part of the edge cloud hosting some network functions such as APs in case of big events. Again, an uninterrupted and low latency communication are critical for these applications to work effectively.
- **Connected vehicle**: Vehicle vendors, network operators and service providers are moving towards smart transport-

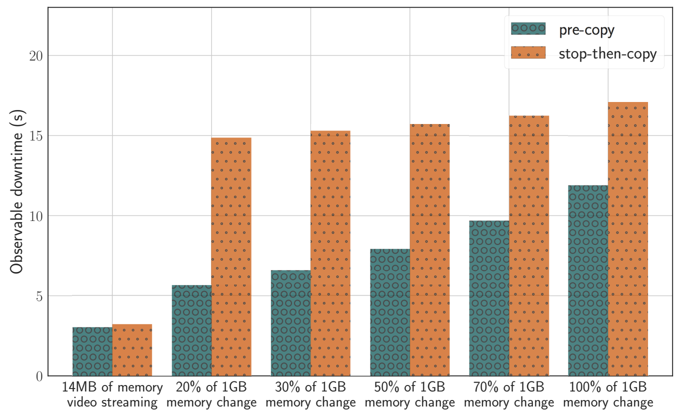

Figure 4: Downtime comparison between pre-copy and stop-then-copy migration of LXC containers.

ation. To enable connected and smart vehicles, massive amount of information, such as position, speed, traffic conditions and road hazards, has to be processed and shared with neighbouring vehicles on the fly. Edge clouds can host applications to provide warnings, driver recommendations and parking slot reservation. These applications require tight collaboration between edge clouds and vehicle, low latency and high computational capacity. Edge application migration of ARNAB can reduce latency and backhaul traffic by migrating applications near to the users.

VI. Conclusions and Future Direction

In this paper, we study the challenges to maintaining the benefits of multi-access edge computing in cloud-enabled WiFi networks. Due to user mobility, network latency increases as users move away from the initial serving edge. Also, in the standard 802.11 protocol, handoff process takes very long which poses an issue for highly interactive and real-time applications. To address these challenges, we proposed ARNAB architecture that employs a double-tier migration for user connectivity and edge applications. Our experimental results show that ARNAB can not only enable continuous service delivery but also outperform the existing work in the area of container live migration across edge clouds. Our future work includes efficient use of storage for container migration by exploiting compression and overlay filesystem. Also, post-copy for container migration can be another extension of this work.

VII. Acknowledgement

This work has been partially supported by the H2020 collaborative Europe/Taiwan research project 5G-CORAL (grant num. 761586).

References

[1] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917–5935, 2017.
[2] "5g-coral H2020 project," 2017.
[3] N. F. V. ETSI, "Network function virtualisation architectural framework," *ETSI GS NFV*, vol. 2, p. v1, 2013.
[4] M. ETSI, "Mobile-edge computing," *Introductory Technical White Paper*, 2014.
[5] A. Mishra, M. Shin, and W. Arbaugh, "An empirical analysis of the IEEE 802.11 MAC layer handoff process," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 93–102, 2003.
[6] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2018.
[7] I. Ramani and S. Savage, "SyncScan: practical fast handoff for 802.11 infrastructure networks," in *24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1, 2005, pp. 675–684.
[8] K. Kwon and C. Lee, "A fast handoff algorithm using intelligent channel scan for IEEE 802.11 WLANs," in *6th IEEE International Conference on advanced Communication Technology.*, vol. 1, 2004, pp. 46–50.
[9] S.-H. Park, H.-S. Kim, C.-S. Park, J.-W. Kim, and S.-J. Ko, "Selective channel scanning for fast handoff in wireless LAN using neighbor graph," in *IFIP International Conference on Personal Wireless Communications*. Springer, 2004, pp. 194–203.
[10] S. Pack, H. Jung, T. Kwon, and Y. Choi, "Snc: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 39–49, 2005.
[11] C.-C. Tseng, K.-H. Chi, M.-D. Hsieh, and H.-H. Chang, "Location-based fast handoff for 802.11 networks," *IEEE Communications letters*, vol. 9, no. 4, pp. 304–306, 2005.
[12] Y. Grunenberger and F. Rousseau, "Virtual access points for transparent mobility in wireless LANs," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2010, pp. 1–6.
[13] M. E. Berezin, F. Rousseau, and A. Duda, "Multichannel virtual access points for seamless handoffs in IEEE 802.11 wireless networks," in *IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5.
[14] J. Schulz-Zander, P. L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz, "Programmatic orchestration of wifi networks," in *USENIX Annual Technical Conference*, 2014, pp. 347–358.
[15] J. Saldana, J. L. de la Cruz, L. Sequeira, J. Fernández-Navajas, and J. Ruiz-Mas, "Can a Wi-Fi WLAN support a first person shooter?" in *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*. IEEE Press, 2015, p. 15.
[16] L. Sequeira, J. L. de la Cruz, J. Ruiz-Mas, J. Saldana, J. Fernandez-Navajas, and J. Almodovar, "Building an SDN enterprise WLAN based on virtual APs," *IEEE Communications Letters*, vol. 21, no. 2, pp. 374–377, 2017.
[17] "Wi-5 H2020 project," 2015.
[18] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam, and M. Rosenblum, "Optimizing the migration of virtual computers," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 377–390, 2002.
[19] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.
[20] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *ACM SIGOPS operating systems review*, vol. 43, no. 3, pp. 14–26, 2009.
[21] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 30, 2014.
[22] J. Nider and M. Rapoport, "Cross-ISA container migration," in *Proceedings of the 9th ACM International on Systems and Storage Conference*. ACM, 2016, p. 24.
[23] R. Morabito, I. Farris, A. Iera, and T. Taleb, "Evaluating performance of containerized IoT services for clustered devices at the network edge," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1019–1030, 2017.
[24] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click modular router," *ACM Transactions on Computer Systems (TOCS)*, vol. 18, no. 3, pp. 263–297, 2000.
[25] F. Fainelli, "The OpenWrt embedded development framework," in *Proceedings of the Free and Open Source Software Developers European Meeting*, 2008.
[26] A. Ceselli, M. Premoli, and S. Secci, "Mobile edge cloud network design optimization," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1818–1831, 2017.