Ultra Reliable Communication for Robot Mobility enabled by SDN Splitting of WiFi Functions

Víctor M. García Martínez, Ricardo C. Mello, Pedro Hasse, Moises R. N. Ribeiro and Magnos Martinello
Federal University of Espirito Santo
victor.manoly@gmail.com,
{ricardo.c.mello;hasse_pedro}@hotmail.com,
moises@ele.ufes.br, magnos@inf.ufes.br

Rafael S. Guimarães
Federal Institute of Education, Science and Technology of Espirito Santo
rafaelg@ifes.edu.br

Valerio Frascolla
Intel Corporation
valerio.frascolla@intel.com

***Abstract*—Wireless networks have become in the last years a key enabling technology for cloud-enabled robots. Among those, the usage of WiFi is a first choice due to its almost ubiquitous use nowadays. However, WiFi suffers from crucial issues like spectrum interference, connectivity losses, long delay for client association and high latency handover. This work proposes a novel architectural split of the WiFi functionalities based on an enhanced software-defined wireless architecture. Cloud-enabled robots scenarios are addressed to derive results showing that the proposed architecture allows uninterrupted communication during handovers, and a quicker failover management.**

***Index Terms*—WiFi, SDN, cloud robotic, URC, mobility**

I. INTRODUCTION

Recent algorithms stemming from cloud computing and other emergent technologies [1], by adding new hash Hardware (HW) requirements, are setting the pace of cloud robotics advancement. On the other hand, thanks to wireless connectivity, robots can access remotely processed information thus mitigating costs and HW complexity, while extending battery life [2]. Hence, reliable and uninterrupted wireless connectivity has become a key feature for robot applications, like remote healthcare and patient locomotion assistance. WiFi is the wireless technology of choice, among the available ones, for many cloud-robotics scenarios, though it suffers from drawbacks like connectivity losses, spectrum interference, long delay for client association and high latency handover, the last two being key hurdles for mobile cloud-robotics deployments.

In WiFi networks working in infrastructure mode, client stations (STA) are associated with an Access Point (AP) to access the network services. Since an STA with one wireless interface can maintain connection with a single AP, if it changes coverage areas a handover mechanism is triggered to keep on communication while changing AP. During the handover, the exchange of management frames between STA and AP introduces a delay for client re-association with the new AP. Additional time is needed for updating ARP tables and for generating new routes so not to lose contact with the moving STA [3]. All that may generate communication interruptions and therefore, in this context, enabling technologies like SW-defined networking (SDN) can be use to obtain ultra-reliable communication (URC).

SDN, a networking paradigm that separates control functions from the data plane, is being enhanced into SW-defined wireless networks (SDWN) [4] by adding clients' seamless mobility and a better quality of service. SDWN makes it possible to create handover mechanisms in which migration decisions are taken by SDN controllers instead of STAs, and allows more effectively update the locations of the clients in the network and the related backhaul routes.

Several studies have been conducted to improve mobility management in SDWN. Ethanol [5], an SDN architecture for WiFi networks, is able to control customer mobility and association processes. In [6] an SDWN is proposed where the creation of virtual AP for each client allows a seamless handover process. In these solutions, although handover is guaranteed without significant throughput degradation, the aspect of updating backhaul after migrations is not addressed. In [7] authors present a mobility management mechanism based on the concept of domains and on updates made in the backhaul, but do not analyze the association delays. In [3] authors propose to use a unique global BSSID for all APs for efficiently addressing the client migration issue, and the delays that the re-association and the backhaul updates introduce, though make its implementation complex, due to the high number of states and the additional synchronization.

Most of those studies have one or more controllers in charge of triggering the handover process and of performing the update of the client's association status in the destination APs. However, none of them analyzes the occurrence of failures in the APs, a phenomenon that would cause a re-association process. In addition, these studies have maintained the operation in the classic infrastructure mode of the WiFi networks, mode that forces clients during their mobility to re-

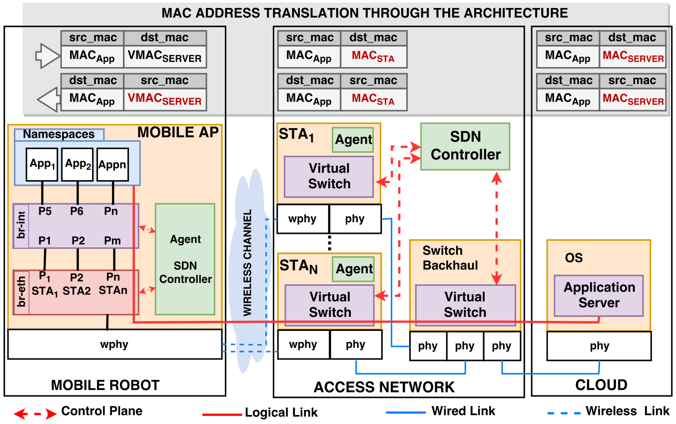

Fig. 1. System Architecture

associate with new APs, thus imposing the need for solutions that minimize the re-association delays. In [8] the mobility is handled by a robot serving as an AP, to obtain the best effective SNR between the AP and the client. Thus, if the normal WiFi infrastructure mode is split into functional blocks, mobility could be placed in the AP and the access to network services could happen through non-mobile STAs.

This work proposes an SDWN architecture for cloud-robotics applications with mobility requirements, which provides seamless connectivity via the splitting of functional elements of conventional WiFi architecture and uses SDN for orchestration. By splitting functionalities, mobility is given to the AP while STAs serve as bridges to interconnect wired and wireless networks. This solution enables multi-connectivity, allowing network multi-path communication, fast handover process and an effective implementation of failover mechanisms. An experiment focusing on a healthcare scenario, in which a cloud-enabled assistive robot provides locomotion services to patients, is performed to validate the proposed architecture in a real environment.

II. ARCHITECTURE

The proposed SDWN architecture aims at achieving a more reliable communication when mobile robots are deployed, for example, in applications involving robotic devices that assist impaired individuals locomotion. The architecture splits the functions in 802.11 network infrastructure creating a MOBILE AP (mAP) located in the robot, and an access network composed by non-mobile STAs that communicates with the cloud as shown in Fig. 1. A centralized SDN controller is in charge of mobility management, operating according to the information of the physical environment provided by the local agents installed in the STAs. In this context, multi-connectivity can be guaranteed even if the STA has just one wireless interface, thus allowing efficient handover processes and failover mechanism to reach ultra reliable communications.

A. Mobile Access Point

In our architecture the mobile node is implemented with a mAP as shown in Fig. 1. The routing mechanism is implemented through two virtual switches, *br-eth* to map the communication between STAs and AP with different ports in the dataplane, and *br-int* to communicate with different applications, each one using its own *namespace*. Rules in *br-eth* are generated using information where each port linked with *br-int* is referenced by AP/MAC and STA/MAC tuples, whereas rules in *br-int* are generated by aggregating a specific server application in one Virtual MAC ($VMAC$).

By construction, *br-int* is instrumental to allow for a proper traffic management, for each flow direction: i) for incoming flows $APP_{server} \rightarrow APP_{AP}$, it installs rules to translate the server application to its corresponding *VMAC*; ii) for outgoing flow $APP_{AP} \rightarrow APP_{server}$, it installs rules which follow the traffic for one or more ports where each of them was previously mapped in one STA. The mAP always sees the $APPserver$ with the same MAC address, reducing the latency generated in handover by capturing the *ARP Request* in the mAP agent, which always replies with its correspondent *VMAC*, allowing each $APPserver$ to make a path decision. Finally, *br-eth* is in charge of: i) forwarding incoming flows to *br-int* bridge using a corresponding mapped port for the source STA; and ii) forwarding outgoing flows changing the MAC_{dst} to the correspondent STA based in the incoming mapping from *br-int*.

B. Access Network

An OpenFlow virtual switch is created in each STA to allow data traffic between the wireless environment and the network. The AP only allows traffic from the MAC address of any of the STAs associated with it and is necessary to implement OpenFlow rules that modify the traffic generated in the cloud making a translation of the source MAC as shown in Fig. 1. The STAs have an agent that sends to the controller the information about the SNR between the STAs and the mAP, in this way the controller can make the handover decisions and implement failover mechanisms.

The backhaul network is composed of an OpenFlow-enabled switch and a centralized OpenFlow controller as shown in Fig. 1. In this part of the architecture lies one of its greatest contributions, since the handover process is triggered only updating OpenFlow rules installed in the backhaul switch, and a synchronization procedure is not necessary anymore. In reality, the mAP does not migrate from one STA to another, it only begins to exchange traffic through another STA. The controller implements failover mechanisms in the same way by updating routes on the switch, once a failure is detected in the STA through which the traffic is passing.

C. Operation

The STAs are configured to make the connection with the wireless network using the mAP's ESSID. The agent installed in each STA sends to the controller information about the SNR of the link between fixed STA's and mAP. In the initial association, the controller selects the fixed STA with the best SNR to attend the traffic to the mAP and installs an OpenFlow rule in the switch to follow the traffic through

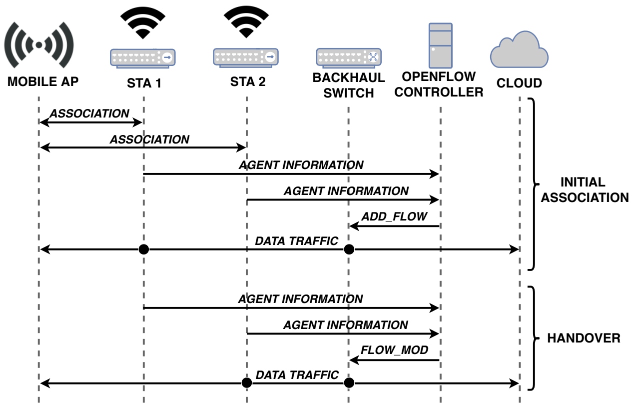

Fig. 2. Operation Process: Initial Association and Handover

the corresponding STA. As shown in Fig. 2, the connection is establishment through STA_1. The agents keep sending SNR status to the controller and if the SNR of any STA associated with the mAP is greater than 3 dB in relation with the actual STA, the controller modifies the flow rule in the backhaul switch to redirect the traffic to this new STA. The route in the backhaul is in charge to archive the best path to the mAP. As shown in Fig. 2 the mobile AP receive traffic from STA_2.

To ensure the operation of the architecture it is necessary that the AP signal provides coverage to more than one STA, so that multi-connectivity can be guaranteed at all times to obtain seamless mobility and virtually zero migration times, thus moving toward a more reliable communication.

III. IMPLEMENTATION

The STAs were implemented in PCs running Ubuntu Server 14.04 on Intel i5-7500 3.40 GHz, 8GB RAM, with an Ethernet interface and a TP-LINK card TL-WDN4800, as a wireless interface capable of working in the 2.4 GHz and the 5 GHz bands. To ensure the operation explained in section II-B, it was installed the Open vSwitch 2.5.2 for OpenFlow 1.3 and an agent developed in python for management of OpenFlow data plane and wireless network information together with the OpenFlow controller. The STAs are connected via Ethernet to an OpenFlow switch Supermicro in the backhaul. This switch connects to a data center where the network services run in a virtualized environment. The implemented controller is a Ubuntu Server 14.04 VM with Ryu 4.18 OpenFlow controller.

The mAP was implemented in a Raspberry 3 model B located in an in-house developed smart walker, a cloud-enabled healthcare robot used for locomotion assistance and rehabilitation [9]. The Raspberry runs the Raspbian Stretch Lite v4.9 operating system on Quad Core 1.2GHz Broadcom BCM2837, 1GB RAM. To provide AP functionalities, it was installed a hostapd 2.5 daemon and a wireless interface TP-LINK TL-WDN3200 (5GHz band) via USB. As with the clients, Open vSwitch 2.5.2 was installed, as well as a local Ryu 4.18 OpenFlow controller responsible for the operation and AP management.

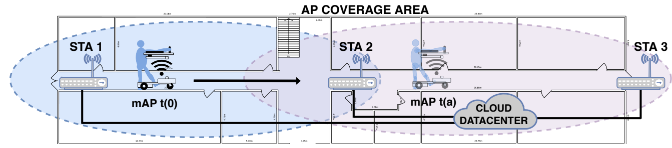

Fig. 3. Experiment Scenario: infrastructure

IV. EXPERIMENTS

The experiment scenario focuses on the use of a smart walker. Nursing homes inhabitants often need some degree of assistance during displacement, and smart walkers are able to provide such assistance while also offering features as obstacle avoidance and guidance, features which are further refined by the integration with the cloud. Due to the inherent mobile nature of such devices, wireless connectivity is necessary and reliability issues pose serious risks to patient safety and integrity. In this scenario, the proposed experiments are built around this need of reliability and seamless handover. The testbest consist of three STAs located in the hall of the CT6 building in Federal University of Espírito Santo as show in Fig. 3, the STAs are connected with a data center, located in the same floor, and are deployed one each 25 meters. In the data center area one can find the backahul switch and the virtualization environment with the OpenFlow controller and the robotic services.

A. Experiment Protocol

The experiments are conducted in the 5 GHz band, to mitigate the interference effects that are common in the 2.4 GHz band. Using the described scenario, the experiments are performed to validate the performance of our proposal. The first test consists of the measurement of the communication throughput during two handover processes that the robot experiences due to its mobility. The robot commanded by its user, moves at approximately 0.5 meters per second, meanwhile TCP traffic is generated between a server in the data center and the robot for 100 seconds, time that allows the robot to move along the entire hall. The second test consists of the same measurements, but increasing the robot's displacement velocity to 0.8 meter per seconds and the duration of the TCP traffic generated to 60 seconds. The main objective of the second test is to observe the failover mechanism of our architecture; for that purpose, the wireless interface in STA_2 is turned off while the traffic to the robot is directed through it.

B. Results and Discussion

Figure 4a shows a multi-connectivity scheme obtained during the first test. At the start, STA_1 and STA_2 are associated with the mAP, while the traffic is routed by STA_1. As the robot moves along the hall, STA_3 receives signal from the AP and is associated with it. Soon after, the first handover occurs and the controller routes the traffic through STA_2. After the second handover, when the traffic is handled by STA_3, the STA_1 client loses signal from the AP while STA_2

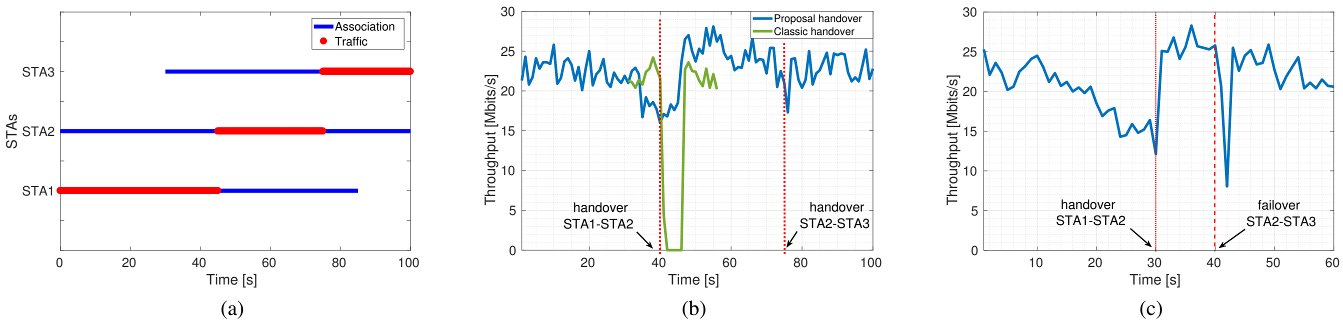

Fig. 4. Experimental Results: a) System Multi-Connectivity; b) Handover of robot mAP; c) Failover of robot mAP.

and STA_3 maintain connection with the AP. It can be observed that the AP always has connection with more than one STA, this feature must be guaranteed during the location of the STAs in the area to be attended. Thanks to the multi-connectivity feature, quick response times are possible during handovers and faults recovery to provide ultra reliable communications.

The results for the handover and failover processes in our architecture show that there are no interruptions in the communication. Figure 4b shows the result of the first test during the two handovers that the robot experiences when passing through $STA_1 - STA_2 - STA_3$. When the handover occurs, despite the related decrease in throughput, it is worth noting that the communication is maintained, in contrast to the communication loss during the classic handover process plotted in green in Fig. 4b. The second test shows similar results during the handover, see Fig. 4c. On the other hand, when a failure occurs in STA_2, the SDN controller routes the traffic towards STA_3. The throughput suffers a greater drop than in the handover since during a failure the controller has to wait for the generation of an unforeseen event by the agent and then make its decision to update the path in the backhaul. Unlike the handover that is implemented as a proactive process, the failover is inevitably reactive, so it takes longer to recover from its effects.

V. CONCLUSIONS

We presented a new architecture to ease reliable communications for robot mobility with SDN orchestration and smart splitting of the functions of the elements of a WiFi network. The use of a mobile AP contributes with multi-connectivity to obtain fast handover and to better handle failovers. The soundness of the architecture is experimentally evaluated, and obtained results satisfy the demands of the cloud robotic applications, providing uninterrupted communication. Future work includes the implementation of multi-path routing mechanisms, and the integration of mobility with programmable residues [10] as enabler for 5G networks. Finally we intend to further elaborate on the results of a recently submitter paper [11] to explore the support of multiple mobile elements.

ACKNOWLEDGMENT

The research leading to these results received funding from the European Commission H2020 programme under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC. In Brazil it has also received funding from CNPq, CAPES and FAPES.

REFERENCES

[1] B. Raaf, M. Faerber, B. Badic, and V. Frascolla, "Key technology advancements driving mobile communications from generation to generation," *Intel Technology Journal*, vol. 18, no. 1, pp. 12–28, 2014.

[2] J. Wan, S. Tang, H. Yan, D. Li, S. Wang, and A. V. Vasilakos, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 4, pp. 2797–2807, 2016.

[3] S. Zehl, A. Zubow, and A. Wolisz, "BIGAP - A seamless handover scheme for high performance enterprise IEEE 802.11 networks," *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, no. Noms, pp. 1015–1016, 2016.

[4] R. D. R. Fontes, M. Mahfoudi, W. Dabbous, T. Turletti, and C. Rothenberg, "How Far Can We Go? Towards Realistic Software-Defined Wireless Networking Experiments," *Computer Journal*, vol. 60, no. 10, pp. 1458–1471, 2017.

[5] H. Moura, G. V. C. Bessa, M. A. M. Vieira, and D. F. Macedo, "Ethanol: Software defined networking for 802.11 Wireless Networks," *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 388–396, 2015.

[6] S. M. M. Gilani, T. Hong, W. Jin, G. Zhao, H. M. Heang, and C. Xu, "Mobility management in IEEE 802.11 WLAN using SDN/NFV technologies," *Eurasip Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.

[7] M. I. Sanchez, A. De La Oliva, and V. Mancuso, "Experimental evaluation of an SDN-based distributed mobility management solution," *2016 ACM Workshop on Mobility in the Evolving Internet Architecture, MobiArch 2016*, vol. 03-07-Octo, pp. 31–36, 2016.

[8] M. Gowda, A. Dhekne, and R. R. Choudhury, "The Case for Robotic Wireless Networks," *Proceedings of the 25th World Wide Web Conference (WWW 2016)*, pp. 1317–1327, 2016.

[9] C. A. Cifuentes, C. Rodriguez, A. Frizera-Neto, T. F. Bastos-Filho, and R. Carelli, "Multimodal Human-Robot Interaction for Walker-Assisted Gait," *IEEE Systems Journal*, vol. 10, no. 3, pp. 933–943, 2016.

[10] M. Martinello, A. B. Liberato, A. F. Beldachi, K. Kondepu, R. L. Gomes, R. Villaca, M. R. N. Ribeiro, Y. Yan, E. Hugues-Salas, and D. Simeonidou, "Programmable residues defined networks for edge data centres," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov 2017, pp. 1–9.

[11] R. Mello, V. M. G. Martinez, R. Guimaraes, P. Hasse, M. R. N. Ribeiro, M. Martinello, and V. Frascolla, "Sdn-based ioiv in future clinical healthcare: Robots for cloud-enabled patient mobility aid," in *IEEE Internet of Things Journal*, 2018 (submitted).