# Building Virtual 802.11 Testbeds Towards Open 5G Experimentation

Kostas Kousias*, Kostas Katsalis†, Donatos Stavropoulos*, Thanasis Korakis*, Leandros Tassiulas‡

*University of Thessaly, Dept. Of Electrical and Computer Engineering, Greece
†Eurecom, Mobile Communications, Biot Sophia-Antipolis, France
‡Yale University, Dept. Of Electrical and Computer Engineering,USA
*{kokousia,dostavro,korakis}@uth.gr,†kostas.katsalis@eurecom.fr ‡leandros.tassiulas@yale.edu

***Abstract*—Together with recent advancements in Radio Access Network (RAN) technologies, Wi-Fi is expected to be at the center of research on the subject of ubiquitous wireless connectivity, towards building the new 5G ecosystem. Nevertheless, the necessary testbed infrastructure to support large scale experimentally driven research seems to be missing. In this work we present the design and implementation of a novel Virtual Wi-Fi Testbed. We present how a traditional Wireless Testbed can support hundreds of virtual Wi-Fi nodes that are open to experimenters. We discuss the design and implementation of the virtualization tools. We demonstrate the accuracy and the overhead analysis of the approach in the face of actual testbed conditions. Implementation experience is also reported on the benefits of using the proposed virtualization approach for a simple association algorithm.**

***Index Terms*—Wireless Network Virtualization, 5G experimentation, Wi-Fi, Testbeds, Multi-SSID**

## I. Introduction

As it is positioned by the large mobile operators and industry players [1],[2], 5G communications will involve a combination of RAN technologies, where a terminal may be connected to several different networks at a given instant. This combination will involve 3GPP technologies (e.g., LTE) as also non-3GPP mechanisms and mainly Wi-Fi technologies. The reason is that Wi-Fi technology is now carrier-grade, while new enhancements in using 802.11ac MIMO technology will increase Wi-Fi speeds to hundred of Mbps, leading to an even better performance than existing 802.11n MIMO. In addition, the widespread deployment of Wi-Fi networks is expected to drive further cellular convergence and the development of innovative services towards 5G communications.

Although the past few years have seen a massive expansion of public and home Wi-Fi installations around the world, the available wireless testbed infrastructures required to support large scale experimentally driven research are missing. The main reason is that in the best case, the number of available nodes on existing testbed installations is limited to up to hundreds. Taking into the account that many users access the testbed resources concurrently, to the best of our knowledge, there is no way and no available open wireless testbed that is able to support multiple concurrent experiments, with dense Wi-Fi installations per experiment.

In this work we present an approach for creating Virtual 802.11(a/b/g/e/n) Access Points (V-APs) in a wireless testbed.

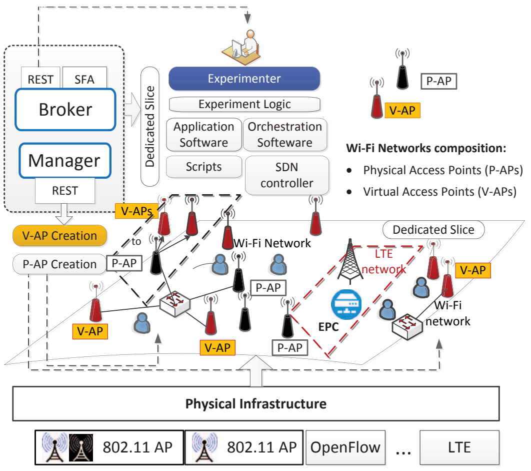


Fig. 1. Network planning in a testbed environment with Physical (P-APs) and Virtual Wi-Fi Access Points (V-APs).

With the devised approach, by using a single testbed node, an experimenter can easily deploy up to a number of virtual access points. Alternatively, instead of reserving a physical node, he can just reserve a number of existing operational V-APs, that will be part of his slice reservation (see Fig.1). The time-consuming process of configuring the node as an Access Point (AP) is avoided and the time for network planning and configuration is dropped dramatically. Thus, on one hand the experimenter is now able to create a cluster of Wi-Fi V-APs with a single REST-POST request, saving configuration effort and time, while letting the focus being on the experiment and the application. On the other hand better utilization of the testbed resources is achieved since clusters of hundreds of V-APs can be created over tens of testbed nodes.

Paper contributions: We present the design of a Virtual Access Points (V-APs) enabled testbed, by employing Multi-SSID services. We have implemented a prototype of the proposed design on a FIRE facility, namely the NITOS wireless testbed [3], located in Volos, Greece. In order to support the new concept, a set of testbed services was created, while the testbed resource advertising & reservation services mechanisms have been extended. These enhancements allow the creation of V-APs through a simple REST interface. We present an overhead analysis of the proposed solution and evaluation results on the effects on throughput performance in the case where multiple flows utilize the V-APs.

We believe our work should shed light on the design space of wireless testbed experimentation. Other testbed operators could use a similar approach to extend the number of available resources and create virtual access points in a similar way VMs are created over existing physical infrastructures.

The rest of the paper is organized as follows. In section II we present the motivation for this work and the current approach on Wi-Fi testbed experimentation. In section III we elaborate the virtualization approach, where in section IV we evaluate the proposed solution, using real exercises on a wireless testbed. We conclude the paper in section V.

## II. Motivation: Experimentally-driven research towards 5G communications

Wi-Fi technologies are going to be increasingly important for both the service providers and mobile operators. The reason is that towards 5G communications, Wi-Fi networks will support many applications which were limited or impossible before, driving new service developments. For example, a number of use cases (e.g., NGMN identified 25 [1]) ranging from Internet of Things (IoT) applications to delay-sensitive video applications, data offloading scenarios, smart cities and so on, require for Wi-Fi support. This list can be quite extensive. Our motivation to extend the testbed capabilities derives from the need to support this type of 5G experimentally-driven research. Nevertheless, the following obstacles currently exist, in cases where multiple Wi-Fi access points are required by multiple experimenters:

- At first there is a strict limit on the available nodes that can be used by the experimenters concurrently, because of the limited testbed resources. For example in Fig. 2 we present a representative daily trajectory of the nodes usage, of the NITOS wireless testbed. The testbed currently offers up to 120 nodes (that can be used as Wi-Fi APs), nevertheless during network courses or because some experimenter reserved many nodes, the possibility of nodes "starvation" is high.

- In the majority of existing testbed infrastructures, the current approach in building Wi-Fi networks is the following. Whenever an experimenter wants a Wi-Fi access point a) he needs to reserve a node (actually only the hardware), b) load an OS (e.g., Ubuntu trusty, windows etc.), c) load the driver (e.g., atheros drivers, iwlwifi) and d) configure the node to operate as an Access Point (802.11a/b/g/e/n). Although this approach gives the experimenter the flexibility to experiment on all the layers of the protocol stack (since he owns the whole resource), it is a procedure that is time consuming, while it needs programming skills and knowledge of the driver operation in order to setup the AP. This complexity can be very restrictive since, not all the experimenters are willing to experiment with the mechanics of the 802.11 networking stack. For example an experimenter may only need some Wi-Fi APs to be used in order to evaluate his off-loading algorithms

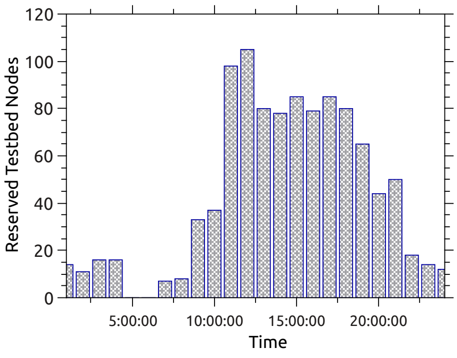


Fig. 2. Nodes usage example in the NITOS Testbed.

for some 5G applications, with the focus been on some other technology e.g., LTE or IoT.

With the proposed approach, multiple virtual access points can be easily created over a single testbed node. This technique could be leveraged by other testbed owners to better serve experimenters who need multiple Wi-Fi APs, without worrying about the configuration details and by bringing the *time to experiment* to few seconds instead of tens of minutes.

### A. Related work on Wireless Testbed Experimentation

Research experimental facilities can be categorized according to their hardware and application domain into Cloud, SDN, Wired, Wireless and Sensor testbeds. In the domain of Cloud experimentation, BonFIRE and Virtual Wall are two indicative testbed representatives, the OFELIA islands are used for OpenFlow/SDN experimentation, while in the field of Wired testbed experimentation Planetlab, Ultra Access, 10G Trace-Tester and PL-LAB, offer pure networking resources to experimenters, like L2 tunnels, NetFPGA cards and programmable networking elements. We note that most of these testbed deployments plan for extensions on the wireless domain due to the importance of research around 5G communications. In the wireless domain several testbeds exist featuring Wi-Fi, LTE, WiMAX, Software Defined Radios (SDR) and Wireless Sensor technologies. Wireless testbed facilities are provided under the GENI initiative in US, with testbeds such as the ORBIT testbed and under the FIRE initiative in EU, with testbeds such as WiLab.t and NITOS. Other prominent wireless testbeds offering Wi-Fi resources, include Norbit and Netmode, while PerformLTE offers LTE experimentation capabilities and FuSeCo is focused on providing 3G and 4G resources. We also mention, LOG-A-TEC that constitutes a cognitive radio testbed and IRIS that provides a significant amount of SDR hardware resources. Finally, SmartSantander and C-Lab are two totally different wireless testbeds, with the first focused on sensors and their wireless protocols, whereas the second forms a community network built and operated by citizens in a decentralized way. In all the aforementioned testbeds the procedure to create Wi-Fi APs requires the time consuming process of actually building and configuring every component of the AP.

## III. Virtualizing a Wireless 802.11 Testbed

Before we proceed with the description of the virtualization approach, a key question we need to answer first is what are we virtualizing in the wireless domain. Or more precisely what we are able to virtualize. Although in the wired world the answer seems obvious (in a switch we can use different flows or different flow space per entity, on server we can use a hypervisor to deploy multiple VMs), in the wireless domain things are more complicated because of the inherent wireless channel sharing operation. For example just building multiple VMs over a single node that carries wireless interfaces, simply is not enough in order to create virtual APs. The same goes for the switching operation; bringing just up multiple virtual wireless interfaces is not enough. As we will explain, special tuning of the driver and special configuration is required in order create multiple V-APs with specific Quality of Service (QoS) characteristics. The things are more complicated when we want to expose this functionality in a testbed environment.

On a basic level, we can use virtualization techniques in the physical layer, the MAC layer, the network layer or even the application layer. We focus on the first three and exploit two models of virtualization [4]: resource based and service based virtualization. In the former model we virtualize a physical resource, (e.g, similarly to a hypervisor that schedules CPU cycles to vCPUs); in the latter we virtualize services (e.g, network services, using L3VPN). When virtualizing wireless access networks, both approaches are valid and the decision clearly depends on the requirements set by the virtual and physical infrastructure owners. For example, if a node carries multiple interfaces or/and multiple antennas, we can share spectrum usage by using multi-user beam-forming or assigning different channels of operation to different interfaces. Then we are talking for resource-based virtualization in the frequency domain. If we want to virtualize a single AP, then Multi-SSID technologies can be applied in order to assign users to different virtual APs [5],[6]. In this case we use service based virtualization, since the same physical resources (frequencies, time domain) are seamlessly controlled and the virtualization takes place in a logical layer.

### A. *Virtual Access Points*

We use the Multi-SSID technique [5],[6],[7],[8],[9] in order to virtualize the testbed nodes and create multiple V-APs. The V-APs emulate the operations of a physical AP at the MAC level. Depending on the driver a different number of V-APs can be supported (e.g., up to 64 V-APs [8][9] or 8 V-APs in ath9k driver). The first thing to clarify, is that essentially a testbed node is a single machine (pc) that carries multiple wireless interfaces and is accessible programmatically using the testbed control and management frameworks. The idea is that in a testbed environment, when you reserve a node you actually "own" the resource and usually through `ssh` access you can perform experiments on the driver, design MAC schedulers, build a wireless topology to run a routing algorithm and so on. Testbed services (e.g. OMF framework [10]) facilitate the loading of the Operating system (OS), the loading of the wireless driver, the network configuration and finally the experiment execution.

Although the Multi-SSID technique is not new, to the best of our knowledge it is the first time that is used in order to use it in a testbed environment. In our case the testbed services were extended in order to support a) the reservation of both virtual and physical resources and b) the creation of the virtual access points over an existing reservation. With the devised approach the experimenter now has the following options in order to provision his network infrastructure: a.

1) Reserve physical nodes and create Physical Access Points (P-APs).
2) Reserve Virtual Access Points (V-APs) that are a-priori build/configured/provided by the testbed.
3) Reserve physical nodes and build V-APs on top. In this scheme the experimenter is able to operate a mixed network with P-APs and V-APs. As we explain in detail in the following, the V-AP creation and configuration is made through a REST interface (see *Manager*, section III-B2).

We explain the procedure of creating the V-APs through a detailed example using the NITOS wireless testbed. We begin by describing the physical underlay, the virtualization approach and then we elaborate the operation of the testbed virtualization services, that are exposed to the experimenters.

***The physical Testbed Nodes:*** The NITOS testbed [3] offers for open experimentation on Wi-Fi, LTE, WiMAX, WSN, USRP, SDN/OpenFlow and cloud technologies. Regarding 802.11 experimentation capabilities, a number of wireless nodes of various types (Icarus, Orbit like, etc.) are available in two open testbeds: an outdoor testbed operating on the roof of a University of Thessaly building (50 nodes) and an indoor testbed operating in a University building basement (60 nodes). For the analysis that follows Icarus indoor testbed nodes were used, each equipped with i7-2600 processor, 8M cache, at 3.40 GHz, 4G DDR3 RAM, Atheros 802.11n PCI/PCI-E chips MIMO, ath9k Linux kernel driver, 1 Gbps Ethernet interfaces (see [3] for details on the testbed hardware specification).

***Building the V-APs:*** The Multi-SSID scheme adopted can be seen in Fig. 3. The setup assumes a linux distribution, wireless interface (we use Atheros 802.11a/b/g/n MIMO) and the ath9k driver. The virtualization approach we will describe is driver depended, but similar approach can be adopted on other driver systems. We plan to extend the virtualization services towards this goal. Deploying the APs requires the configuration of the `hostapd.conf` file and enabling the `hostapd` service. The `iw` command (e.g., `iw list`) can be used in order to discover the capabilities supported by the wireless card and furthermore the number of SSIDs that can be supported. In the `hostapd.conf` file we are able to configure various parameters per V-AP like: the SSID, the bridge name, the authentication/authorization/encryption and the QoS characteristics).

***Bridging the interfaces:*** In the heart of the proposed approach resides Linux bridging. The reason is that we wanted for every V-AP to use a different VLAN in order to identify

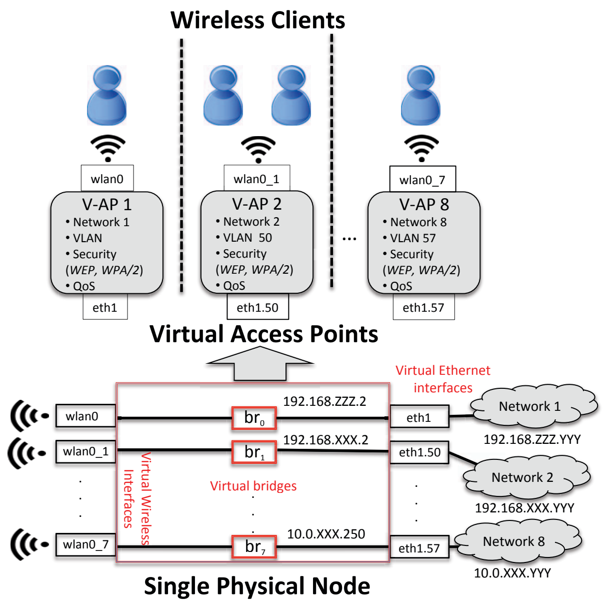


Fig. 3. Virtual Wi-Fi Access Points.

the different flows. Since in the native `hostapd` operation a single bridge can be created, we created manually the additional bridges and added the corresponding additional wireless interfaces in the `/etc/network/interfaces` file. For every bridge added, a series of configuration commands lines were required (e.g., `iface brName inet manual`, `bridge_port wlan0_0` etc.). This created the bridges and added a virtual wireless interface to it. The Ethernet interface could be added manually afterwards using the `brctl` tool. As last step, one can assign IP addresses to the bridge interfaces. Every IP range of every interface should belong to different subnet in order to separate the incoming traffic.

***Security considerations:*** Regarding encryption schemes, the Multi-SSID technique supports a different scheme per V-AP, while a different authentication/authorization mechanism is also supported. Note that in a testbed environment the experimenters can use multiple testbed resources and resources advertised through testbed federations. In order to support the new concept, we extended the resource advertisement, reservation, authentication/authorization mechanisms of the testbed. This procedure is described in detail in the following (see the *Broker Service* description, section III-B1 ). We note that currently for simplicity reasons we do not use RADIUS authentication. RADIUS services integration to the existing testbed AAA mechanism is planned for future work.

***QoS considerations:*** In the wired world not only virtualization of the network resources but also the ways to achieve service guarantees and QoS are straightforward. In contrast, in the wireless domain there are many limitations imposed by the stochastic nature of the wireless channel and the CSMA/CA operation. Recent research works in 802.11 virtualization rely on the Beacon messages to adjust per-station minimum contention windows and transmit limits to affect the airtime usage to different group of users [11]. In fact, the `hostapd` file configuration allows for different 802.11e configuration per V-AP (with different Contention Window characteristics per traffic/queue type). This way we are able to change priorities and provide different QoS per V-AP. Note that because of CSMA/CA operation, there exists a fair share in the time domain of airtime usage (in the long run), for every client independently of how many V-APs operate. This greatly affects the overall performance. In [12] a service differentiation scheme is presented using programmable routers in 802.11 APs and a feedback-based mechanism to guarantee specific throughput ratios between competing flows. We plan to extend the QoS mechanism per V-AP and provide an additional service for the V-AP's QoS, using alternatives designs where programmable data-planes like OpenVSwitch or the Click router are utilized instead of Linux bridging.

## *B. Reservation, Control& Management of Virtual Resources*

A number of enhancements was necessary in the control and management frameworks of the NITOS testbed, in order to support the concept of virtual 802.11 access points. These are related with the extension of the *Broker Service* and the design and implementation of the new *Manager Service*. A high level description of their components can be seen in Fig.4. We note that in the NITOS testbed two widely used frameworks were already utilized for the control and management of the testbed infrastructure. These are the OMF Framework, regarding nodes management and control and Slice-based Federation Architecture (SFA) interface, regarding resource abstraction and reservation (see Fed4FIRE [13] for details on these interfaces). The extended *Broker* wraps around SFA, while the *Manager* utilizes various OMF services.

*1) The Broker Services:* The *Broker* service [13] is responsible for the resource advertising and the resource reservation, while it is the component that controls the slicing of the resources and guarantees slices isolation. It is able to programmatically interact with other resource management systems (e.g. OpenNaaS) or GUI based reservation systems, in order to reserve the physical or virtual network elements. In more detail, it keeps an inventory with information regarding all the available resources and their virtualization capabilities, which are then exposed through a new REST interface. As a pilot case in the NITOS testbed a set of 16 V-APs are operational and exposed through the *Broker* service. With the extended *Broker* services we give to the experimenter (or to remote management systems) the ability to reserve both physical nodes and V-APs. In the case of a physical node reservation the experimenter needs to configure the node as an AP, while in the case of a V-AP reservation the access point is pre-configured. In the case physical nodes are reserved, the experimenter can easily create virtual 802.11 resources using the *Manager Services*. With these services, he is able to configure information like the BSSID, VLAN tagging, IP network etc. per V-AP.

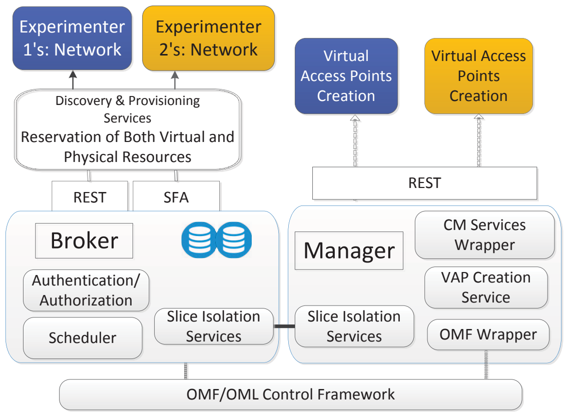

Experimenter 1's: Network
Experimenter 2's: Network
Discovery & Provisioning Services
Reservation of Both Virtual and Physical Resources
REST
SFA
Broker
Authentication/ Authorization
Scheduler
Slice Isolation Services
Virtual Access Points Creation
Virtual Access Points Creation
REST
Manager
CM Services Wrapper
VAP Creation Service
Slice Isolation Services
OMF Wrapper
OMF/OML Control Framework

Fig. 4. Testbed Virtualization Services.

**VAP 1**

```
{
      "id":"0",
      "ssid":"test-ssid-A",
      "password":"test-a",
      "max-customers":"20",
      "network":"192.168.3.0",
      "netmask":"255.255.255.0",
      "vlan":"50",
      "max-rate":"1024",
      "min-rate":"1",
      "ratio-rate": "0.2",
      "beacon_int":"100",
      "max_num_sta":"255",
      "wmm_enabled": "1",
      "ieee80211n":"1",
      "ht_capab":"[HT40+]..."
},
```

**VAP2**

```
{
      "id":"1",
      "ssid":"test-ssid-B",
      "password":"test-b",
      "max-customers":"10",
      "network":"192.168.2.0",
      "netmask":"255.255.255.0",
      "vlan":"692",
      "max-rate":"1024",
      "min-rate":"1",
      "ratio-rate": "0.2",
      "beacon_int":"100",
      "max_num_sta":"255",
      "wmm_enabled": "1",
      "ieee80211n":"1",
      "ht_capab":"...[DSSS_CCK-40]"
      "wmm_ac_bk_cwmin":"4",
      "wmm_ac_bk_cwmax":"10",
      "wmm_ac_bk_aifs":"7",
      "wmm_ac_bk_txop_limit":"0",
      "wmm_ac_bk_acm":"0",
      "wmm_ac_be_aifs":"3",
            ...
}
```

Fig. 5. V-AP configuration examples.

***The scheduler:*** The Scheduler component is in charge of resolving any conflicting reservation requests that arrive from both interfaces and acts also as a policy enforcement point regarding resource usage prioritization.

***SICS:*** The *Broker* exposes a *Slice Isolation and Control Service* (SICS) to the *Manager*, to infer if the requested actions to the physical resources should be authorized or not.

***REST & SFA:*** Next to the REST interface the Broker features a SFA interface [13], which is the de-facto framework for testbed federation (used for example in the Fed4FIRE federation [10]). Through these interfaces it accepts requests for resource reservations, thus keeping each time the availability and the ownership status of every wireless resource. It is important to note that the REST API is working side by side with the SFA interface. This is achieved by having integrated these interfaces to the *Broker* and sharing a common inventory in the back-end. Thus, every change in the inventory as a consequence of a REST or an SFA call is reflected immediately, regardless the interface that is being used. Note that the SFA interface has already a predefined schema which we didn't want to modify, as it is used by the federation tools and services.

*2) The Manager Services:* A secured REST interface (x509 certificates and HTTPS) is exposed that is responsible for the management and control services. In our design and implementation the *Manager* operation relies on remote *ssh-based* access to the nodes or to the testbed's control systems.

***OMF & CM Wrapper:*** The OMF control framework [10] is utilized in order to perform actions like "OS image load" on the reserved physical nodes, using the OMF/PXE service. A wrapper is also build over the CM services. The CM service is used to perform actions like *Turn Node ON*, *Turn Node OFF* or check *Node Status*. The functionality of the OMF and the CM services is now exposed by the *Manager's* REST interface.

***P-AP and V-AP configuration:***

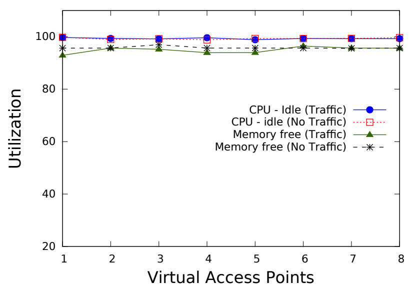
(P-APs): Through the Manager's REST interface, the experimenter is now able to "transform" the node to an Access Point, by configuring all the network interfaces, all the `hostapd` relevant information (channel of operation, SSID, QoS, MIMO operation etc.) and bringing the `hostapd` service up, all through a single POST request. This gives the experimenter the ability to start the experiment execution in seconds, since no scripts writing is required and no detailed knowledge of the `hostapd` service internals.

(V-APs): through a single POST request with a JSON body, using the ath9k driver, the experimenter is able to create up to 8 V-APs over a single node. A sample configuration per V-AP is presented in Fig. 5. For both V-APs, information like the SSID, beacon-interval, max-number of associated users, network and VLAN information, etc. are configured. In addition, regarding QoS characteristics, in V-AP 1 the default configuration is loaded, where for V-AP 2, different Contention Window parameters can be configured per traffic/queue type on both the uplink and downlink. Note that depending on the data-plane utilized, additional functionalities can be described like different *ratio-rate* per V-AP (meaning how much percent of the actual throughput to guarantee per V-AP). This functionality will be supported in the next manager release, with the addition of programmable data-planes like OVS or the Click router to the Multi-SSID scheme described.

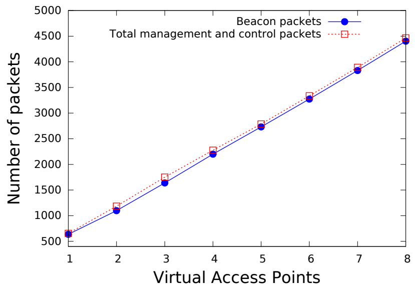
## IV. Evaluation

Although the Multi-SSID technique is widely used, especially in cases where different security considerations or/and different QoS needs exist, it suffers from increased overhead by means of bandwidth utilization. The reason is that every V-AP uses a significant fraction of the total bandwidth for its management traffic [8],[9]. Thus its use is avoided in experiments where maximum performance is the objective. Nevertheless, with proper tuning, it can satisfy some minimum performance thresholds in order support dense Wi-Fi networks for multiple experiments.

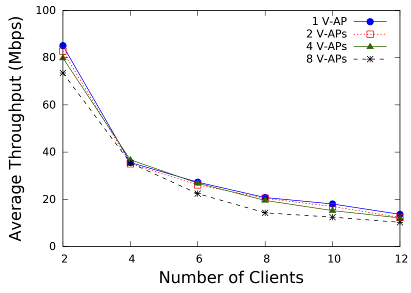
In this section, we demonstrate the wireless virtualization system efficiency, by means of system CPU and memory utilization as well as throughput performance. Further network overhead issues accounting the virtualization mechanism adopted, are also presented. The NITOS indoor wireless testbed (Icarus indoor nodes)[3] was exclusively used in order



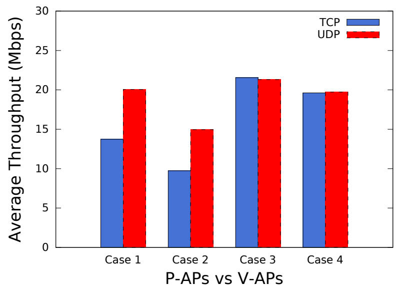
(a) CPU and MEMORY utilization.



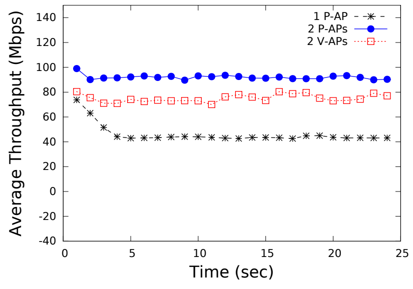
(b) Control and management overhead.



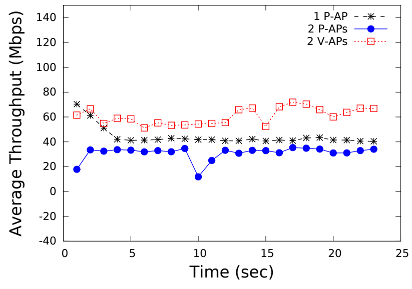
(c) The effect of increasing the number of clients in average throughput performance



(d) (C1):8 P-APs-same channel, (C2):8 P-APs-different channel,(C3): 1 P-AP, (C4): 8 V-APs



(e) Association Alg.: Flow A performance by means of average throughput for all flow A's clients.



(f) Association Alg.: Flow B performance by means of average throughput for all flow B's clients.

Fig. 6. Evaluation of the V-AP approach in the NITOS wireless testbed.

to perform the following series of experiments (see III-A for the nodes' specification). We note that all the experiments where conducted under real interfering conditions, since other experiments were running in parallel in the testbed. This affected the overall throughput performance.

In Fig. 6(a) we demonstrate the CPU and memory utilization for the case of increasing the number of V-APs. At each time each V-AP was serving a different client. In y-axis, we depict a) the average percentage of CPU that remains idle during the experiment and b) the unexploited RAM memory. Note that the case of 1 V-AP is the case of a single AP operating over the node. We made the tests for two different traffic scenarios: a) the clients are just connected with the V-APs without receiving or transmitting any traffic and b) all the clients interact concurrently with their V-APs sending TCP traffic to some external server (`iperf` tool). As we can see, the addition of V-APs has no effect on the CPU or memory of the system in any of the two cases. Similarly, memory hardly overpasses the threshold of 95% during all the tests. These results were expected, since in the case of simple packet forwarding operations by the network card, not all the protocol stack is called.

Multiple factors affect the actual throughput performance like the number of connected stations, the number of V-APs, the number of neighboring APs that operate in the coverage area of the AP, while also channel conditions and RSSI levels. Thus multiple experiments were contacted in order to have a safe conclusion. As it was also expected theoretically, because of the V-APs operation there is a decrease in performance due to increased management and control overhead. Because of V-APs operation there is an increase of the beacon messages and the management probe requests/responses. Beacons are management frames that send from a wireless interface periodically in order to inform the nearby users of their existence. These phenomena can be observed in Fig.6(b) and 6(c). As we can see in Fig.6(b) this increase is linear to the number of V-APs, while increasing the number of clients in Fig. 6(c), results in an average throughput decrease per client. in this case again the worst performance can be observed in the case of using 8 V-APs, due to the increased overhead. This increase can greatly affect the overall performance and depending on configuration like the beacon interval, can significantly vary.

Indeed, in Fig. 6(d) we present the comparison for the average throughput achieved by 8 clients that were connected to P-APs or V-APS as follows:

-*Case 1:* each client was connected to a single P-AP, where all P-APs were transmitting in the same channel.

-*Case 2:* each client was connected to a single P-AP, where all P-APs were transmitting in different channels (nearby overlapping).

-*Case 3:* all the clients are connected to a single P-AP.

-*Case 4:* each client is connected to single V-AP, where all the V-APs were hosted in the same physical node.

The topology of the nodes was the same during all the experiments in order to receive more reliable comparison (keep the clients to AP distance the same).

Again, we made the experiments using the *iperf* tool and provide both TCP and UDP results (we present the average performance a user enjoyed sampled over 10 experiments). We see that highest average throughput is achieved in cases 3 and 4, where we operate the single P-AP and the multiple V-APs over a single physical node. A throughput decrease is also

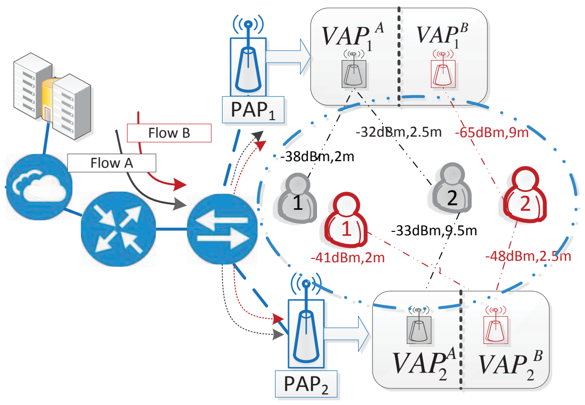


Fig. 7. Association algorithm performance using V-APs.

observed between these two cases because of the overhead that beacon messages cause in the case of V-APs.

## A. V-APs under different association patterns

Furthermore, we implemented an indicative scenario leveraging the proposed virtualization scheme, to investigate the performance of a simple association scheme. Similarly, Multi-SSID have been used in [14] for trajectory mining. A trajectory consists of a time series of sensor readings of all Wi-Fi signals measured by a device, regarding signal strength. This information can be used to design different association policies.

In Fig.7 we assume a system with two nodes, two service providers $A$ (BSSID A) and $B$ (BSSID B) and a number of four users all in the same geographical area. We assume that two users are associated with provider $A$ and two users are be associated with provider $B$. Each node can serve either as a P-AP or it can host multiple V-APs. Note that each client has different distance from both the physical nodes.

**Setup 1:** Two Icarus nodes operate as 802.11n Access Points (AP) operate without virtualization capabilities and each serves two users. **Setup 2:** In the second setup, at each node two V-APs operate. One V-AP serves $A$'s users and the other one serves $B$'s users. In this case, the algorithm decides association with the V-AP on a signal strength/distance basis (e.g. we associate the second user with the closest AP).

In Figs. 6(e) and 6(f), we present the average throughput performance, for provider's $A$ clients (flow A) and provider's $B$ clients respectively (flow B). In all cases constant TCP traffic was generated using the *iperf* tool. As we can see, in the case where we use two 2 V-APs, there is a slight decrease in performance ($\sim$ 20Mbps) for flow A in comparison to the case of using a single P-AP to serve flow A. Nevertheless, this decrease is counterbalanced equivalent by a throughput increase in flow B. This happens since the topology is different with clients having closer distance to the V-APs (and thus the physical nodes), thereby experiencing similar RSSIs. This way the sharing of the medium happens in a way where the two flows are now more balanced. A balanced behavior is also depicted in the case of using a single AP to serve both flows, nevertheless the average throughput in this case is less when compared to case of using 2 nodes and 2 V-APs per node.

## V. Conclusions & Future Work

Towards 5G communications, Wi-Fi networks will be increasingly important, since they are expected to support many applications and new service developments. In this work, we demonstrated a simple way for provisioning V-APs in a wireless testbed, where the number of physical resources are limited. Our experiments present that the addition of V-APs do not affect the CPU and memory of the physical host, while it can have negative effect on the throughput performance since the management overhead is increased. Nevertheless, this approach allows for experimentation on multiple algorithms and scenarios, without worrying about the nodes starvation or configuration cost when deploying hundreds of Wi-Fi APs in a testbed environment. Future plans include the support of programmable data-planes like OVS and Click modular router with the Multi-SSID scheme and investigation of virtualization mechanisms for future Wi-Fi 802.11ac networks.

## Acknowledgements

This work was carried out with the support of the Fed4FIRE project, an Integrated project receiving funding from the European Union's FP7 programme, under grant agreement no 318389. Kostas Katsalis work has been supported by the 5G-PPP project "Coordinated control and spectrum management for 5G heterogeneous radio access networks" (COHERENT).

## References

[1] N. Alliance, "5g white paper," *Next Generation Mobile Networks, White paper*, 2015.
[2] "Wireless Broadband Alliance." [Online]. Available: http://www.wballiance.com/
[3] "NITOS Testbed." [Online]. Available: http://nitlab.inf.uth.gr/NITlab/
[4] N. M. K. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *Communications Magazine, IEEE*, vol. 47, no. 7, pp. 20–26, 2009.
[5] C. Xiang, "Access control method for wifi device and wifi device," Jun. 19 2014, uS Patent App. 14/145,234. [Online]. Available: https://www.google.com/patents/US20140169354
[6] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri, "Splitap: Leveraging wireless network virtualization for flexible sharing of wlans," in *IEEE, GLOBECOM*, Dec 2010, pp. 1–6.
[7] G. Aljabari and E. Eren, "Virtual wlan: extension of wireless networking into virtualized environments," *International Journal of Computing*, vol. 10, no. 4, pp. 322–329, 2011.
[8] "Cisco ios software configuration guide for cisco aironet access points," *Cisco Press,Cisco Systems, Inc.*, 2010.
[9] "Improve air quality by minimizing ssids: Using role-based access to increase wi-fi application performance," *White paper,Aruba Networks*, 2010.
[10] "Fed4FIRE Project." [Online]. Available: http://www.fed4fire.eu/
[11] K. Guo, S. Sanadhya, and T. Woo, "ViFi: virtualizing WLAN using commodity hardware," in *Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*. ACM, 2014, pp. 25–30.
[12] K. Katsalis, V. Choumas, K. T., and L. Tassiulas, "Virtual 802.11 Wireless Networks with Guaranteed Throughout Sharing," *ISCC, IEEE*, 2015.
[13] D. Stavropoulos, A. Dadoukis, T. Rakotoarivelo, M. Ott, T. Korakis, and L. Tassiulas, "Design, Architecture and Implementation of a Resource Discovery, Reservation and Provisioning Framework for Testbeds." WiNMeE, 2015.
[14] M. Werner, L. Schauer, and A. Scharf, "Reliable trajectory classification using wi-fi signal strength in indoor scenarios," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*. IEEE, 2014, pp. 663–670.