Addressing the *eduroam* inter-domain mobility problem through virtual APs

Silvia Surroca*†, Daniel Camps-Mur*, Ilker Demirkol†
*i2CAT Foundation, Barcelona, Spain
† Universitat Politecnica de Catalunya, Barcelona, Spain
{silvia.surroca, daniel.camps}@i2cat.net, ilker.demirkol@entel.upc.edu

***Abstract*—The Education Roaming (*eduroam*) is a widely spread federation of Wi-Fi networks that allows users to freely roam between providers. The key to eduroam's massive adoption has been the fact that a single global network identifier is used across all the federated Wi-Fi networks. However, when multiple eduroam networks are deployed in an overlapping geographical area, the current design may cause client devices to inadvertently roam between providers, hence potentially disrupting applications and hindering network planning. In this paper we study how virtual Access Points (vAPs) can be used to address the inter-domain mobility problem in eduroam. Our main contribution is a novel mechanism to steer clients across vAPs, which mitigates unintended handovers. The proposed extensions have been validated experimentally considering a wide range of commercial Wi-Fi clients available in the market.**

I. Introduction

The *eduroam* initiative is a federation of Wi-Fi networks that has achieved a remarkable penetration throughout the world, especially across academic institutions. Thousands of institutions from 80 different countries offer *eduroam* services nowadays, resulting in billions of *eduroam* authentications per year [1].

A key requirement to be part of the *eduroam* federation is to deploy a Wi-Fi network where the Access Points (APs) advertise *eduroam* as their *Service Set Identifier (SSID)*. The SSID is the network identifier of a Wi-Fi network and is advertised within the Beacon frames, transmitted by the APs typically every 100 ms. Thus, the connection manager of a Wi-Fi client uses the SSIDs contained in the received Beacon frames to look up the credentials required to join a known network in an automatic manner. In the case of *eduroam*, WPA2 authentication is used and hence the required credentials consist of a client certificate issued by the origin or home institution of the user. In order to be able to authenticate roaming users, for example a student from a German university accessing an *eduroam* network in Barcelona, the hierarchy of RADIUS servers depicted in Fig. 1 is used. Individual institutions offering *eduroam* services connect their RADIUS server to a RADIUS server managed by the NREN[1] of their country. Consequently NRENs from different countries are connected through a RADIUS server hosted by GEANT[2]. The

[1]NRENs are the operators of the National Research and Education Networks. For example the Spanish NREN is Rediris.

[2]GEANT is a project gathering all European NRENs to foster joint projects and infrastructures.

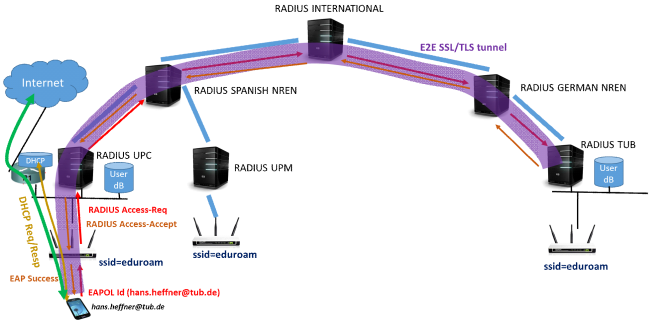


Fig. 1: *eduroam* architecture.

key to route an authentication request through the hierarchy of RADIUS servers is the *eduroam realm*, which is declared by the requesting device in the EAP messages depicted in Fig. 1. Notice that the hierarchical RADIUS network depicted in Fig. 1 is used only for authentication purposes, the data plane (user traffic) is always routed through the visiting institution. This means that roaming *eduroam* users do not have access to the internal services offered by their home institution unless they use additional means, e.g. a VPN.

While the decision of using the same SSID in all *eduroam* Wi-Fi networks simplifies roaming, it introduces a problem when multiple overlapping *eduroam* providers coexist in the same geographical area. This problem is commonly known within the *eduroam* community as the *Russell Square* problem [2], because it is specially relevant around Russell square in London where one can find many institutions offering *eduroam*. The Russell Square problem manifests when Wi-Fi devices connected to their home *eduroam* network, autonomously roam to an overlapping *eduroam* network. The cause of this problem is that network selection in Wi-Fi devices is largely based on signal strength (RSSI), which can heavily fluctuate among networks present in an overlapping area. Indeed, current Wi-Fi chipsets cannot distinguish Beacons from different *eduroam* providers because they all advertise the same SSID, namely *eduroam*. The consequences of the Russell Square problem are manyfold: i) from the client perspective, active sessions may be disrupted because IP addresses are not maintained when roaming to a different network, or suddenly the user may not be able to reach the internal services of his home institution; ii) from the network administrator perspective, the Russell Square prob-

lem complicates network planning decisions, as an institution offering *eduroam* may suddenly receive a surge of traffic from clients belonging to overlapping *eduroam* networks, which may hinder the QoS of the home users of that institution.

The main contribution of this paper is a novel Wi-Fi access architecture, which uses virtual Access Points (vAPs) and L3 tunnels between overlapping *eduroam* providers to mitigate the Russell Square problem. The core concept is a novel mechanism to steer Wi-Fi clients to the appropriate vAP. The proposed user steering mechanism has been validated experimentally with a wide set of Wi-Fi clients. A second contribution of this paper is a characterization of the Wi-Fi association behavior of the analyzed Wi-Fi clients, which may lead to the definition of novel load balancing or network selection algorithms beyond those presented in this paper.

This paper is organized as follows. Section II surveys current approaches to the Russell Square problem, as well as relevant work on virtual APs and user load balancing. Section III introduces our system architecture and novel user steering mechanism. Section IV provides an experimental performance evaluation of our architecture, and characterizes the network selection behavior of a wide sample of Wi-Fi clients. Finally Section V concludes the paper.

II. Related Work

There are several solutions proposed by the *eduroam* community to mitigate the effects of the Russell square problem [2], although no solution is considered perfect. The preferred solution is to move to an infrastructure supporting 802.11u [3], also known commercially as *Wi-Fi Passpoint*. Passpoint allows an AP to be identified not only through the SSID, but also by means of additional signaling elements conveying the home service provider and its roaming partners. Thus, in a setting with overlapping *eduroam* networks, an 802.11u client can be configured to always prefer the network of its home institution. The problem faced by service providers moving to 802.11u is that the level of support among clients is poor, and when supported, interoperability issues still arise. Thus, although promising, a solution based on 802.11u appears to still be some years down the road. Other recommended methods consist of conducting wireless coverage surveys to minimize overlapping areas, or simply to avoid providing *eduroam* services when a neighboring institution is already doing so. These two solutions though face clear problems in terms of complexity and fairness. The most commonly used mitigation method nowadays consist of sharing L2 VLANs between neighboring institutions. Following this method each service provider reserves a VLAN per neighboring provider to route users back to their home institution, which is possible because RADIUS servers allow to bind *eduroam* realms to L2 VLANs. Notice that this mechanism though is only possible when neighboring providers share a common L2 infrastructure offering VLAN services. In addition, when adopting this solution, service providers cannot police the wireless bandwidth used by roaming users, and thus their home users may still suffer from degraded QoS. In this paper we propose a novel scheme, inspired on this technique, which alleviates the previous two issues.

The *eduroam* architecture introduced in this paper is based on the concept of virtual Access Points (*vAPs*). The technology required to support vAPs, also known as multiple SSIDs, is widely available in enterprise Wi-Fi products [4], and consists of having a single physical AP advertise multiple Beacon frames with different BSSIDs[3], which are perceived by the client devices as coming from different physical APs. This technology has been used to improve station mobility in [5], by moving vAPs between physical APs, instead of forcing the client device to handover. Commercial products have also adopted vAPs as a means to provide per client QoS, since clients need to adhere to the QoS settings advertised by the AP they are associated to [6]. In this paper, we use vAPs to enable QoS bandwidth provisioning but not on a per user basis, which has been proven not to scale [7], but rather on a per overlapping *eduroam* provider basis.

Finally, relevant to our work are network controlled load balancing mechanisms for Wi-Fi enterprise networks. In [8], a mechanism is introduced that allows the network to handover devices between APs with delays below 100 ms. This solution though requires a set of overlapping APs to advertise the same BSSID, and it only works for devices operating at the 5GHz band. In [9], an enterpise Wi-Fi system is proposed where client load balancing is achieved by having the APs forward Probe Requests from the client devices to a centralized controller, which then configures the AP the client needs to associate to respond to that Probe Request. This solution though is not applicable to the Russell square problem for the following reasons. First, overlapping APs belong to different service providers, and are therefore not connected to the same controller. Second, given the large quantities of Wi-Fi devices in current networks, processing Probe Requests in a centralized manner is not scalable. In the next section, we introduce a novel mechanism to balance users between vAPs that does not achieve the real time performance of [8], but works with commercial Wi-Fi clients, both at 2.4 GHz and 5 GHz, scales to large device densities, and requires only a software upgrade to existent Wi-Fi APs.

III. System design

The main concept behind our proposed scheme is to instantiate in each physical AP a set of vAPs representing the overlapping *eduroam* providers. For example, in a setting where three providers overlap, each physical AP instantiates three vAPs[4]. In addition, each vAP is bound to an IP tunnel[5] that connects a given vAP to the home network of its corresponding *eduroam* realm. The goal of our architecture is that although clients belonging to a given *eduroam* realm may connect to any physical AP, they should always access the

[3]In Wi-Fi, the Base Station ID (BSSID) is the MAC address of the AP.

[4]For reasons detailed later we instantiate $n+1$ vAPs, where n is the number of overlapping providers.

[5]IP tunnels, provisioned a priori between *eduroam* providers, are used to facilitate connectivity even when providers do not share an L2 network.

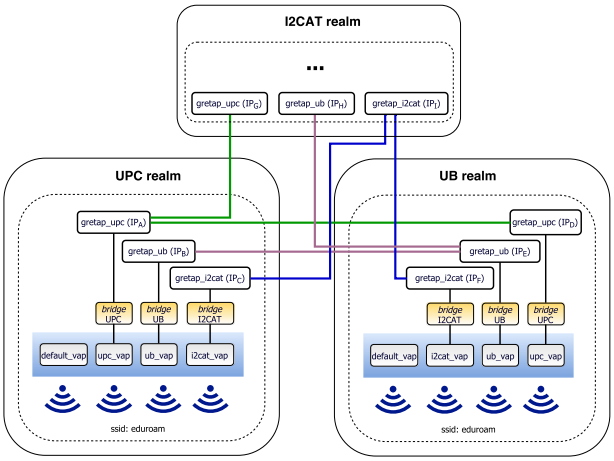


Fig. 2: Proposed *eduroam* access architecture.

network through the vAP representing their institution, hence traffic will be tunneled back to their home network, and the client will be oblivious to the Russell square problem. Our architecture is depicted in Fig. 2 for an example where we have three overlapping providers: UPC, UB, i2CAT.

There are two main issues that need to be resolved in this architecture. The first one relates to scalability concerns, both in terms of vAPs and IP tunnels. The second one relates to how to force a client to connect through a given vAP, since AP selection logic is not standardized.

Regarding scalability, it is worth noticing that in practice we expect a limited number of overlapping *eduroam* providers in a geographical area, e.g. < 5. In this regime, we consider the overhead involved in pre-instantiating vAPs for each overlapping provider, and pre-instantiating point to point IP tunnels to be manageable. If the number of overlapping providers were to grow beyond those limits, then more efficient techniques would be required. For example, the over the air overhead introduced by vAPs can be reduced using some of the mechanisms standardized in 802.11v or 802.11u; however these mechanisms are not readily available in many commercial devices. Regarding IP tunnel overhead, it is easy to see that the number of tunnels grows as $n(n-1)$, where n is the number of overlapping providers. To alleviate this fact one could define a neutral point, e.g. a *mobility exchange* (MEx) point to provide routing service between tunnels, such that each provider only needs to maintain a single tunnel with the MEx. As previously stated though, we do not expect scalability to be a problem in practice. Regarding the specific IP tunnel technology, we have decided to use EoGRE[6] because it is readily available in many Wi-Fi enterprise products.

Regarding the problem of how to force clients to associate through the vAP representing their institution, notice that clients will receive Beacon frames from all vAPs with SSID set to *eduroam*. In this setting we expect terminals to randomly choose the vAP to connect to; hence, a mechanism is needed to guide the terminals to the appropriate vAP. In our design we use for this purpose white and black lists of

[6]Ethernet over Generic Routing Encapsulation is a tunneling protocol that emulates a LAN between remote peers.

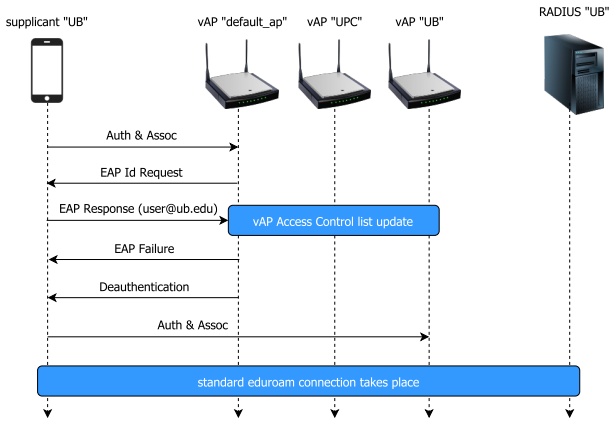


Fig. 3: Assignment of vAP in a terminal's first connection.

MAC addresses associated with each vAP. In particular, a vAP can have a whitelist (accept MAC list), which only allows the terminals that are in the list to connect, or a blacklist (deny MAC list), which accepts all the terminals that are not listed. The challenge however, is that the system cannot know a priori the MAC addresses of the terminals that could connect from each *eduroam* realm, and hence the MAC accept lists cannot be pre-configured. To address this issue we define another vAP (namely *default_vap*) configured to accept access requests from terminals connecting for the first time (through a blacklist).

Once a terminal connects for the first time, *default_vap* learns the *eduroam* realm of the user and the client's MAC address and configures the per-realm MAC access lists accordingly. To do this, *default_vap* uses the user name provided by the terminal during the *eduroam* authentication process. Finally, once the MAC access lists are configured, *default_vap* sends a disassociation message to the terminal, to force the new client to reconnect through the vAP representing its realm. This process is depicted in Fig. 3. In Section IV we will evaluate the effectiveness of this scheme in forcing clients to connect through the correct vAP.

We conclude the description of our scheme discussing several aspects. First, *eduroam* users not belonging to any of the realms represented by the vAPs, e.g. foreign students, can still connect through our system as they do in standard *eduroam*, for example through *default_vap*. Second, instantiating per-realm vAPs does not only solve the mobility problems that appear in the Russell square scenario but also provides new tools for policing wireless bandwidth. In particular, the Wi-Fi channel access settings can be configured for each vAP in order to provide over the air prioritized QoS to the users of each realm [10]. Finally, for scalability reasons MAC lists should be kept small (remove stale entries). However, our evaluations show that very long lists (1000s of entries) can be processed in less than a second by simple Raspberry Pi devices.

IV. Implementation and Experimental Evaluation

A. Prototype description

To implement the proposed solution, first an AP Manager module has been developed that offers a REST-based API

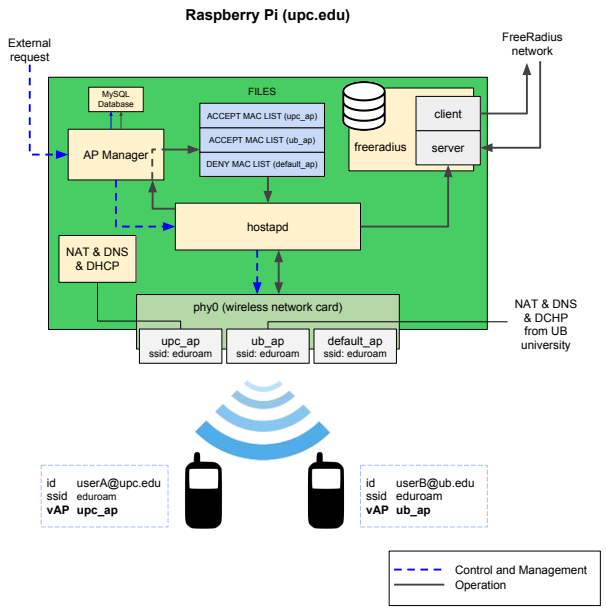


Fig. 4: Implementation of an *eduroam* provider on our testbed.

to instantiate and configure vAPs. Then, we use *hostapd*[7] to manage vAPs, and to provide *eduroam* authentication services. In our implementation we have modified *hostapd* in order to capture the binding between a client's MAC address, and its *eduroam* domain, and to configure the per-vAP access lists accordingly through the AP Manager module. We use *freeradius*[8] to implement the RADIUS-based authentication scheme of *eduroam*. Fig. 4 depicts the software architecture that we use to represent an *eduroam* provider in our testbed, which we physically deploy on a Raspberry Pi (RPi). Notice that RPis are chosen due to their reduced costs, however in a practical setting our system can be deployed in any bare metal AP supporting a Linux based OS.

Out testbed consists of two RPi devices representing two overlapping *eduroam* providers, which we refer to as the *UPC* realm and the *UB* realm. Then, a third RPi device is used representing a router that provides IP and RADIUS connectivity between the two providers. As illustrated Fig. 4 each *eduroam* provider instantiates three vAPs, namely one vAP per provider and the *default_vap*. Finally, the corresponding EoGRE tunnels are also set up. For our experiments, we use 2 overlapping providers, with 3 vAPs each (including *default_vap*).The results presented in this section should scale linearly with the number of vAPs, although as discussed in Section III we do not expect a high number of vAPs in practice.

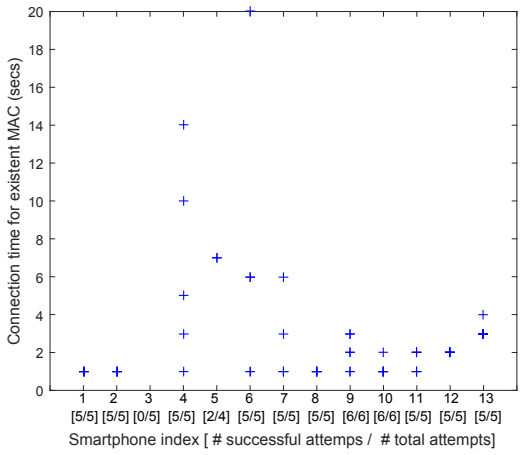
B. Evaluating WiFi client steering

In order to evaluate the effectiveness of the proposed client steering mechanism with commercial Wi-Fi devices, we consider two situations: i) the behavior of a Wi-Fi client with a MAC address that is already known by the system, and ii) the behavior of a Wi-Fi client with a MAC that is not known to the system. For these two cases we perform a set of tests with the 13 commercial smartphones depicted in Table I configured with *eduroam* credentials from one of the two realms considered in our prototype. The selected smartphones correspond to the maximum number of smartphones available through colleagues in our lab, and although a limited sample we argue that it is representative of users in student environment, thus relevant to evaluate *eduroam* services. For each test we measure the time that each smartphone needs to associate with the vAP serving its *eduroam* realm. The results are reported in Fig. 5.

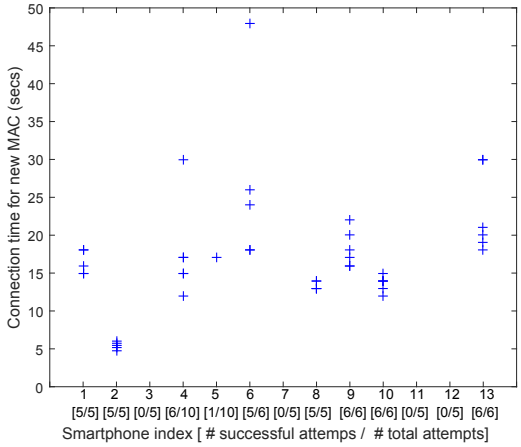
We start discussing in Fig. 5(a) the case where the MAC of the requesting Wi-Fi client is already known to the system. Being the MAC known to the system means that the access lists in *default_vap* and the per-realm vAPs are already configured. However, this does not mean that the client will directly access the system, since the client receives a set of Beacon frames with SSID set to *eduroam* and different BSSIDs, and the AP selection logic is client dependent and not standardized. In particular, in order to be compatible with our scheme we expect clients to select a vAP at random, try to authenticate, and after receiving an authentication failure if the selected vAP is not correct, continue trying with the subsequent vAPs until the correct one is found. Looking at Fig. 5(a) we observe two types of client behavior: i) Random vAP selection, as we were expecting, is observed in 11 out of 13 terminals, and ii) sending the authentication request always to the vAP with smaller MAC address (BSSID), which is the case for the Jiayu G4 and the BQ Aquaris A4.5 clients. The latter behavior is not compatible with our approach because such terminals may never connect to the right vAP. However, we consider this behavior a bad and not representative implementation in general, not only for our scheme, since the vAP with the lowest MAC might be faulty or would be overloaded through this behavior. We further classify the behavior of clients that select all vAPs at random into: i) clients that when rejected by one vAP try out the next vAP with a short delay, this is the majority of terminals that end up experiencing a network entry delay below 4 seconds, ii) clients that stick to the first vAP they choose with a higher probability, e.g. the Samsung Galaxy Mini and the Jiayu G3, which results in higher delays, and iii) clients that introduce a significant delay before retrying with a new vAP after being rejected. In the worst case the experienced network entry delay is 20 seconds. Given that network entry in Wi-Fi smartphones is often performed in background, we argue that these results, although coming from a limited sample, hold promise into the practical applicability of the proposed scheme.

Next we analyze in Fig. 5(b) the time it takes for a terminal to connect to the correct vAP when the client's MAC is not known to the system. This case is different than the previous one in that the client is accepted to $default\_vap$, then the EAP authentication procedure is initiated, and once

[7]https://wireless.wiki.kernel.org/en/users/documentation/hostapd
[8]http://freeradius.org/



(a) Connection time for a known client MAC



(b) Connection time for an unknown client MAC

Fig. 5: Client steering mechanism results (x axis index corresponds to smartphone number in Table I).

TABLE I: Smartphones Evaluated for the Client Steering Mechanism

Device #	Device model	OS version	Timeout(s)
1	BQ Aquaris E5	Android 4.4.4	10
2	iPhone 4	iOS 7.2.3	2
3	Jiayu G4	Android 4.4.2	∞
4	Samsung Galaxy Mini	Android 2.3.6	2
5	BQ Aquaris A4.5	Android 5.1.1	∞
6	Jiayu G3	Android 4.2.3	12
7	One Plus ONE	Android 5.1.1	∞
8	Motorola Moto G	Android 6.0	7
9	LG Nexus 5	Android 6.0.1	13
10	Samsung Galaxy Grand 2	Android 4.4.2	12
11	iPhone 6	iOS 7.1.2	∞
12	Samsung Galaxy Tab S	Android 5.0.2	∞
13	Samsung Galaxy S6	Android 6.0.1	16

the AP discovers the user's realm, a Deauthentication message is sent, the MAC address lists in the vAPs are configured, and we are back to the case where the MAC is known to the system. However, the behavior of a client that receives a Deauthentication message is again client dependent.

Looking at Fig. 5(b) we found two types of behavior: i) Reconnection attempt after a certain timeout, which is specified in Table I, and ii) No reconnection within 60 seconds, which is specified as ∞ in Table I. The latter case is obviously bad for our scheme, but it is worth noting that what these results imply is that the terminal does not reconnect automatically, but as we explain next the user can always connect manually.

1) Discussion: Looking at the results in Fig. 5 it is important that a practical system attempts as much as possible to have the user MACs registered in the system. In enterprise networks APs are often managed by a controller and therefore it is easy to maintain per-vAP centralized MAC access lists accessible to all APs. As explained in Section III there should be no scalability issues in centralizing MAC access lists. In addition, to facilitate inter-domain handovers, we recommend that a Wi-Fi controller exposes an API to allow an overlapping provider to update its MAC access list once a new client is detected; we have built this functionality into our prototype. Despite the previous optimizations, it is unavoidable to have cases where clients will connect to the system for the first time. We see this as a manageable problem, since, even for the non-compliant[9] terminals identified in Table I, a manual reconnection triggered by the user will succeed, because the MAC address is always registered after the first connection attempt. For the compliant terminals in Table I, even if the network entry times in Fig. 5(b) may be high, in practice clients connect without user intervention, so an increased network entry time does not directly impact user experience. Finally, an optimization for future work is to identify the WiFi manufacturer by MAC address in *default_vap*; and if an unsuitable chipset is found, a standard *eduroam* connection could be provided [12].

C. Complete network entry delay

This section compares the durations of a conventional *eduroam* connection and a connection made with our improvement implemented in the network. Tests are performed with an iPhone 4, which as depicted in Fig. 5 is one of the clients with an implementation with low connection delays. In addition, to simulate physical distances between the two institutions (namely a UB user connecting at UPC) we introduce an artificial delay of 100 ms, which we consider a worst case in practical situations, since overlapping *eduroam* providers are typically located in the same city. Thus, the results of these experiments capture both the network entry delay, and the delay introduced by the EoGRE tunnels.

Fig. 6 depicts the results obtained in 15 connections for both the regular *eduroam* network setup and our scheme. In Fig. 6, two components of the connection time are plotted in the x and y axes. First, *EAP completion time* measures the time since the terminal sends the first authentication request until the correct vAP transmits an EAP Success message. Second, *DHCP Time*

[9]Devices with $Timeout = \infty$ in Table I

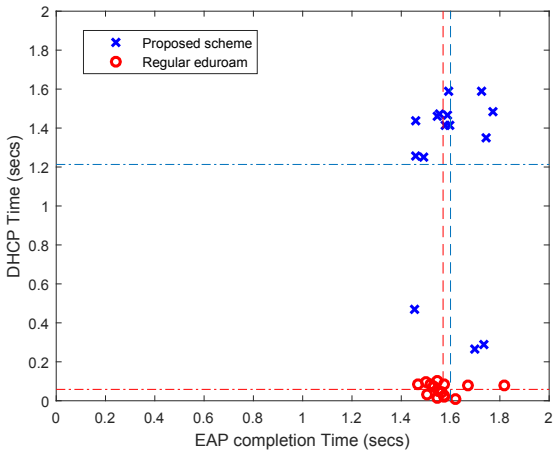


Fig. 6: DHCP vs EAP connection durations in current *eduroam* configuration (blue) and the proposed scheme (red) for 15 trials. Dashed lines represent average connection times.

captures the time between receiving the EAP Success message and obtaining a valid IP address.

We can see in Fig. 6 that *EAP time* is almost the same for standard *eduroam* and for our scheme. The reason is that as explained in the previous section the iPhone 4 quickly attempts to reconnect until it finds the appropriate vAP resulting in a negligible delay increase with respect to standard *eduroam* setup. Then, once the connection to the appropriate vAP is established, in both cases RADIUS packets travel to the home domain, which in this case dominates the overall delay. On the other hand, as expected, the time the terminal takes to obtain a valid IP address increases in our scheme, because the DHCP requests travel back to the home DHCP server. The overall connection delay in our scheme stays below 2 seconds, which we consider acceptable given the benefits obtained by solving the Russell square problem. Nevertheless, 100ms of per-hop delay is a pessimistic estimate since the institutions located in the same city and connected by an education network, in practice, will have much lower values.

D. Handover performance

Finally, we evaluate an inter-domain handover scenario where a terminal moves between vAPs of different institutions, i.e. overlapping *eduroam* providers. We use our proposed enhancement of propagating MAC lists between overlapping providers, so that the receiving provider already has the per-vAP access lists configured before the handover. However, when a terminal does a handover between institutions, it must perform a full *eduroam* authentication and request again an IP address from the home DHCP server, which in our case is going to provide the terminal with the same IP it had before the handover. We emulate the handover scenario by physically removing the Wi-Fi adapter from the Raspberry Pi a terminal is connected to, to force the terminal to connect to the other institution. After 15 experiments with the iPhone 4, we found that the terminal reconnects to the next AP (Raspberry Pi) after 4 to 6 seconds, with an average of around 5 seconds. Although the handover is not seamless, we have verified that services using application level buffering like Youtube are not interrupted during the handover [11].

V. Conclusions

The Russell square problem can affect the quality of experience of end users, and difficult network planning decisions when multiple providers offering an *eduroam* service overlap in the same geographical area. With the technologies available in the market today, only complex inter-provider agreements can be sought to try to mitigate this problem, which often cannot be implemented in practice. In this paper we have proposed a novel *eduroam* access architecture that addresses this problem and can be deployed with technologies readily available in the market. We have evaluated our architecture using an experimental testbed, and a wide set of commercial devices. In addition, through our study we have performed a characterization of the network selection procedures of a smartphone sample comprising some of the main vendors in this space, which has the potential to fuel new research in the area of network controlled load balancing in Wi-Fi.

VI. Acknowledgements

The research leading to these results has received funding from the EU FP7 Programme (FP7/2007-2013) under REA grant agreement n 618098, and from the Spanish Ministry of Economy and Competitiveness (MINECO), under research grant TEC2013-47960-C4-4-P and RYC-2013-13029.

References

[1] Eduroam adoption, available online, https://www.eduroam.org/where/
[2] Eduroam FAQ, available at: https://community.jisc.ac.uk/library/janet-services-documentation/faqs-eduroam-system-administrators-and-implementation-techs-0
[3] Hiertz, Guido R., Dee Denteneer, Lothar Stibor, Yunpeng Zang, Xavier Prez Costa, and Bernhard Walke, "The IEEE 802.11 universe," IEEE Communications Magazine 48, no. 1 (2010): 62-70.
[4] Aruba Networks, The Impact of Multiple SSIDs on Wi-Fi Performance, available online: http://community.arubanetworks.com/t5/Community-Tribal-Knowledge-Base/The-Impact-of-Multiple-SSIDs-on-Wi-Fi-Performance/ta-p/25374
[5] Yan Grunenberger and Franck Rousseau, "Virtual access points for transparent mobility in wireless LANs" In Wireless Communications and Networking Conference (WCNC), 2010 IEEE, pages 16. IEEE, 2010.
[6] Trudeau, Pierre, and Stephane Laroche, "Configurable quality-of-service support per virtual access point (VAP) in a wireless LAN (WLAN) access device," U.S. Patent No. 8,885,539. 11 Nov. 2014.
[7] Zdarsky, Frank A., Ivan Martinovic, and Jens B. Schmitt, "The case for virtualized wireless access networks." In Self-Organizing Systems," pp. 90-104. Springer Berlin Heidelberg, 2006.
[8] Zehl, Sven, Anatolij Zubow, and Adam Wolisz, "BIGAPA seamless handover scheme for high performance enterprise IEEE 802.11 networks," Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP. IEEE, 2016.
[9] Murty, R., Padhye, J., Chandra, R., Wolman, A. and Zill, B., "Designing High Performance Enterprise Wi-Fi Networks" In NSDI (Vol. 8, pp. 73-88), 2008, April.
[10] Prez-Costa, Xavier, and Daniel Camps-Mur, "IEEE 802.11e QoS and power saving features overview and analysis of combined performance," IEEE Wireless Communications 17.4 (2010): 88-96.
[11] S. Surroca, D. Camps-Mur, I. Demirkol, "i2CAT/UPC - vAPs for eduroam". Available online: https://www.youtube.com/watch?v=rdfhALtvIkk
[12] Gentry, D., Pennarun, A. (2016). Passive Taxonomy of Wifi Clients using MLME Frame Contents. arXiv preprint arXiv:1608.01725.