NxWLAN: Towards Transparent and Secure Usage of Neighbors‘ Access Points in Residential WLANs

Piotr Gawłowicz, Sven Zehl, Anatolij Zubow and Adam Wolisz
{gawlowicz, zehl, zubow, wolisz}@tkn.tu-berlin.de
Telecommunication Networks Group, Technische Universität Berlin, Germany

***Abstract*—The increased popularity of IEEE 802.11 LANs (WLANs) in residential (home) environments leads to dense, unplanned and thus chaotic deployments. Access Points (APs) are frequently deployed unfavorably in terms of radio coverage and use interfering frequency/power settings. In fact, in some parts of a one's apartment the usage of the neighbor's AP might be preferred as it might provide better signal quality than the own AP. Moreover, the network performance could be dramatically improved by balancing the network load over spatially co-located neighbor APs operating on different radio channels or having asymmetric workloads.**

We address these problems by presenting NxWLAN (Neighborhood extensible WLAN) which enables the secure virtual extension of user's home WLANs in residential environments through enabling the usage of neighboring APs operating on the same or on different radio channels. NxWLAN requires no additional software to be installed on the client STAs while providing the same level of security as the home AP, e.g. WPA2, without revealing any kind of security credentials to untrusted neighboring APs.

In this paper we prove, by prototyping and simulating the NxWLAN solution, that it is operational, performant and easy to deploy using off-the-shelf hardware. Moreover, we provide the full source code of our prototype to the community as open-source.

***Index terms*—** Wireless, WLAN, Residential Wi-Fi, AP Virtualization, Wi-Fi Sharing

I. Introduction

In recent years we have seen a rapid growth of IEEE 802.11 wireless LAN (WLAN) usage in residential networks. According to Cisco Visual Networking Index (VNI) [1] by the end of 2019 more than half of the worldwide IP traffic will be generated by WLAN devices. Moreover, according to Watkins et al. [2] already in 2015 nearly 70 percent of all households (worldwide) equipped with broadband Internet access own at least one WLAN Access Point (AP). These two trends lead to dense AP deployments in both urban and suburban residential areas.

Shi et al. [3] evaluated the potential for mutual WLAN sharing (i.e. allowing others the access to ones AP) in residential environments and revealed that in sparsely-populated suburban areas mutual WLAN sharing can be advantageous. Specifically, the authors revealed that for about 15% of clients, their home APs do not provide the best signal for more than 50% of time. Additionally, due to their smaller coverage areas residential APs have unsynchronized workloads [4], [5], [6] and hence there is an opportunity to utilize unused resources (both radio and backhaul) of neighboring APs by means of balancing client stations.

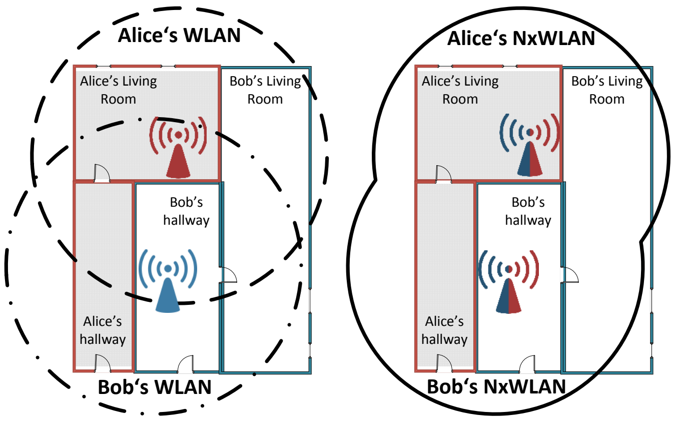

Fig. 1. Vision of shared APs usage: classical WLAN (left) and NxWLAN (right).

Unfortunately as by today the residential APs are under the individual management of their owners, and while even in one‘s apartment usually tens of neighboring APs are visible [7] – only the own one might be used.

WLAN-sharing is prohibited due to following reasons:

i) lack of trust in the guest's scope of actions, e.g. Internet piracy – restrictive regulations like the *"Störerhaftung"* in Germany [8] makes users responsible for all traffic that goes over their home AP,
ii) fear of performance degradation due to consumption of additional radio and backhaul resources,
iii) overhead in and side effects of configuration for admitting third party users,
iv) expectation of unfair behavior of neighbors, i.e. free-riders,
vi) suspicion of the owner's disrespect for privacy, e.g. recording of user information.

In this paper we present **NxWLAN** (Neighborhood extensible WLAN), which enables the virtual extension of WLAN through secure usage of neighboring APs, cf. Fig. 1. The NxWLAN approach achieves this by deploying a Wireless Termination Endpoint (WTP) on each neighboring AP and by tunneling encrypted 802.11 traffic to the Virtual AP (VAP) residing on the home Enhanced AP (EAP). This allows a client device to always authenticate against the home EAP using the WPA-PSK pass-phrase already stored in the device, without revealing the password to the – possibly untrusted – visited AP of the neighbor.

There is no need for a registration process or software to be installed on the user's devices. The visited EAP can enforce its own strategy (policy) as for the amount of resources shared with the visiting clients. Last but not least while accessing the Internet via neighbors' EAP the user retains the possibility of accessing all of her home LAN devices (e.g. printer, appliances) in the usual way.

The main contributions of this paper are:

i) the design of NxWLAN architecture, with use of the modern solutions derived from the research on enterprise and residential WLAN,
ii) a prototypical implementation of NxWLAN using off-the-shelf hardware and open source software
iii) a performance evaluation using a small 802.11 indoor testbed and network simulations in ns-3.

The NxWLAN prototype is provided as open source: **https://github.com/nxwlan**

II. NxWLAN's Design

A. Requirements

The main goal of NxWLAN is to open any home AP to be used by users of neighboring APs. We believe that the proposed solution should meet the following requirements:

- Possibility to use the neighbors‘ APs to access the home network and the Internet without disclosing the own access credentials for the WPA-PSK method of securing the own AP access,
- Assuring that only neighbors having verified privileges to access own AP will be admitted by the neighbors' AP,
- Support for steering client stations to the most suitable AP for means of providing the best possible radio link quality and network load in the wireless channel of home and neighboring AP as well as available Internet backbone capacity, i.e. maximizing both network and user performance,
- Support for sharing policy configuration – the load created by the visiting STAs should not create unacceptable burden on the usage of the AP by its owner,
- Providing roaming STA with link layer connection to home WLAN (layer 2) – users should be able to access all devices, including printers, NAS, etc.,
- Lack of any specific, additional (re)configuration or software installation at a client station (STA) – it would be inconvenient for users and almost impossible to provide software for all kind of STA devices with plethora of operating systems, wireless NIC, etc.,
- Simple software installation process at AP and no need for reconfiguration nor registration of home/visiting STAs – residential WLAN are usually managed by people lacking experience and knowledge of wireless networking.

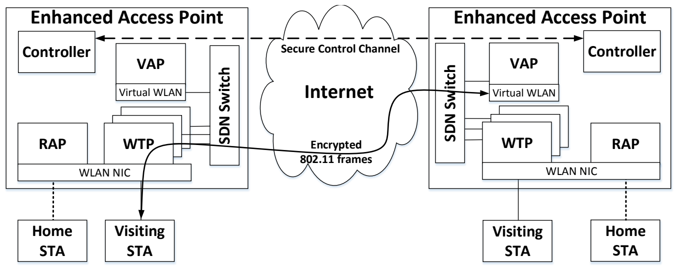

Fig. 2. Architecture of residential NxWLAN network.

B. Architecture Overview

The overview of the NxWLAN architecture is presented in Fig. 2. In order to support NxWLAN, each AP has to be upgraded with some additional software components to become an Enhanced Access Point (EAP). Specifically, each EAP consists of a Real Access Point (RAP), a Virtual Access Point (VAP), a set of Wireless Termination Points (WTPs), a SDN switch and a controller.

The RAP is the legacy AP that is regularly used by STAs in absence of cooperative neighbors. A single VAP is activated on-demand in the home EAP when discovering at least one NxWLAN-enabled neighboring EAP willing to share its wireless and backhaul resources with visiting client stations. The VAP takes care of the generation of all 802.11 management and data frames. It is responsible for encryption and decryption of 802.11 data frames. Moreover, it is in charge of authentication and association of STAs that are connected via the WTP residing on the visited EAPs. Note that the VAP uses the same configuration as the RAP, so a STA can connect to it without any reconfiguration.

The WTP is a simple proxy (radio head), residing on the visited EAP that works as follows: i) it transmits encrypted 802.11 frames, received (over a secured Internet tunnel), from the VAP residing on the home EAP, to STAs and ii) it forwards encrypted 802.11 frames, received from STAs (via the wireless channel), to the VAP in the home EAP (over a secured Internet tunnel). An EAP may host multiple WTPs, i.e. one WTP is deployed on-demand in the visiting EAP for each neighboring NxWLAN-enabled EAP.

The VAPs and the WTPs are interconnected using Internet network tunnels. Note, that a single VAP can be connected with multiple WTPs and it may serve STAs connected over different WTPs. To this end, the SDN switch is used to properly steer and forward traffic received from WTP to the corresponding VAP and vice versa.

The controller creates and manages all entities present in an EAP and communicates with other controllers residing in neighboring EAPs in a secure way over encrypted IP tunnels.

C. NxWLAN Setup Procedure

In order to bootstrap NxWLAN, first neighboring EAPs have to discover each other. To this end, we adopted the over-the-air discovery procedure provided by the ResFi framework [9]. It provides an EAP with an interface allowing it to announce its public IP address and additional security credentials to neighboring EAPs. Moreover, the interface also enables to receive in return the public IP address and the security credentials of these neighboring EAPs. These credentials are then used to setup bi-directional secure communication tunnels over the Internet. ResFi makes use of additional vendor specific information elements added within *Probe Response* (PRES) and *Probe Request* (PREQ) frames to enable these data exchanges during the standard 802.11 active scanning procedure. This way all neighboring EAPs (even those operating on different channels) are able to discover each other.

Fig. 3 illustrates the NxWLAN Setup Procedure. Upon the discovery, the EAP1 creates a VAP and sends a *WTP Setup Request* messages to all discovered neighboring EAPs. According to the configured policies, the neighboring EAPs may reject or proceed this request. After a successful creation of the WTP and network tunnel, the EAP2 responds with a *WTP Setup Complete* message to the requesting EAP1. In case of rejection, e.g. due to being overloaded by large number of visiting STAs, an EAP may not respond. Upon the reception of the confirmation message, the EAP1 completes the VAP configuration and sets up a tunnel according to the configuration received from the neighboring EAP2. From this point, the VAP and WTPs are able to exchange (encrypted) 802.11 frames over the tunnel.

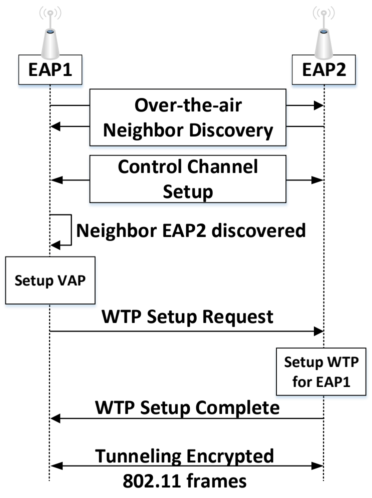

Fig. 3. EAP Neighbor Discovery Procedure and Setup of WTP and VAP.

D. STA Association Procedure

Fig. 4 presents a STA association procedure in NxWLAN. In order to connect to any EAP, an 802.11 STA performs normally an active scan by broadcasting a *PREQ* sequentially over all available WLAN channels as specified in the 802.11 standard. The *PREQ* can be received potentially by a RAP directly or indirectly by VAPs using the corresponding WTPs deployed on the visiting EAPs. In standard WLANs, all APs that are in reception range of the STA transmitting a *PREQ*, should send a unicast *PRES* immediately upon the reception of a PREQ as a STA stays on single channel only for limited amount of time [10]. In NxWLAN, the RAP answers as usual, while all WTPs forward the received PREQ over tunnels to the corresponding VAP, which in turn generates a *PRES* to be tunneled back to the corresponding WTP for transmission.

In NxWLAN, we exploit the locality of the target environment, i.e. the fact that there is a limited number of neighboring EAPs, and we therefore simply forward the *PREQ* to every neighboring VAP, i.e. participating EAPs. Note, thanks to this mechanism NxWLAN works also when MAC address randomization is used in client stations, e.g. Android OS since version 6.0 [11].

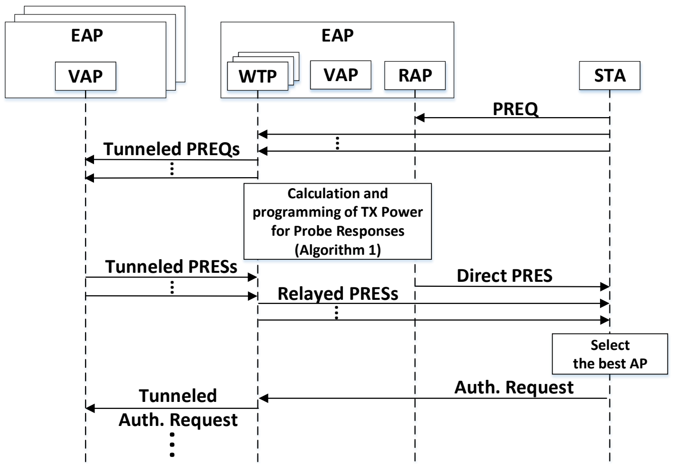

Fig. 4. Client association process in NxWLAN. For reasons of clarity, the reception of the PREQ by only a single visiting EAP is shown.

E. STA Steering

For network load (in radio and backhaul) balancing reasons NxWLAN steers a new associating client STA to a particular EAP to be either directly or indirectly served by the Home AP (RAP) or any of Neighbours' APs (WTP) respectively. Therefore, on receiving a *PREQ* frame each EAP estimates its willingness to serve this client directly via RAP or indirectly via one of the deployed WTPs. The willingness depends on the EAP‘s available capacity in the radio and backhaul network as well as the quality of the radio link towards the STA. Note, that the computation of the willingness is different for RAPs and WTPs as for the latter ones the network traffic has to be tunneled over the backhaul to the corresponding VAP. Hence, also the available backhaul capacity at the EAP housing the VAP needs to be taken into account. To this end, each EAP periodically measures and announces its available DL and UL bandwidth to its peers.

Unfortunately, due to tight timing constraints during search phase, i.e. a STA stays on a distinct channel only up to 100 ms [10], it is not possible for the EAPs to harmonize their willingness values among themselves in order to reach consensus on the best EAP to serve that particular STA. In NxWLAN we solve this as follows. We exploit the fact that an ordinary 802.11 STA always connects to the AP that provides the strongest signal, i.e. as measured from the *PRES* frame. Therefore, each EAP encodes its willingness by means of setting a proper transmission power for the *PRES* frames to be sent to the STA, i.e. in general different values for RAP and WTPs. If the transmit power is selected properly, it is possible to steer the client STA to an intended RAP or WTP of a particular EAP. Therefore, we have to assure that the received power on the client side is higher for the *PRES* sent by the EAP with higher willingness. We have designed and implemented an algorithm for computing the TX power, that takes the available capacity in both the wireless and the wired network (backhaul) as well as the quality of the radio link towards the STA into account (Algorithm 1). Note, that each EAP is able to compute willingness of serving the new STA of all neighboring EAPs having only their DL/UL capacity reports as input.

Moreover, our algorithm requires knowledge of the pathloss between STA and EAP. Here we assume the transmission power of the STA is set to a fixed value, does not change over time and is known to all EAPs. As future work, we would like to utilize the 802.11k and 802.11v standards that allow STAs to report to EAP the RSSI of received beacon frames.

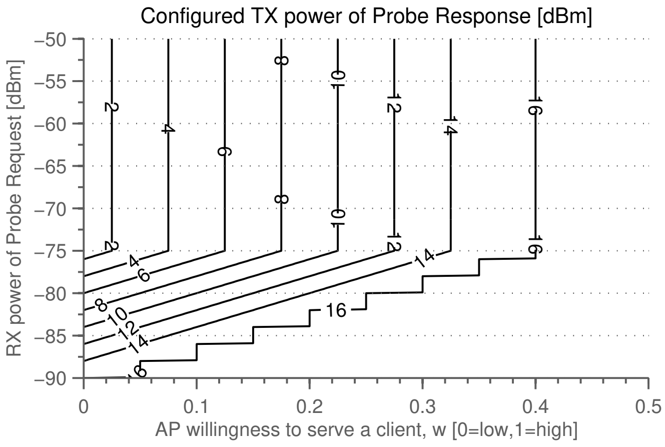

Fig. 5. EAP encodes its willingness to serve a client in the transmit power of *Probe Response* packet.

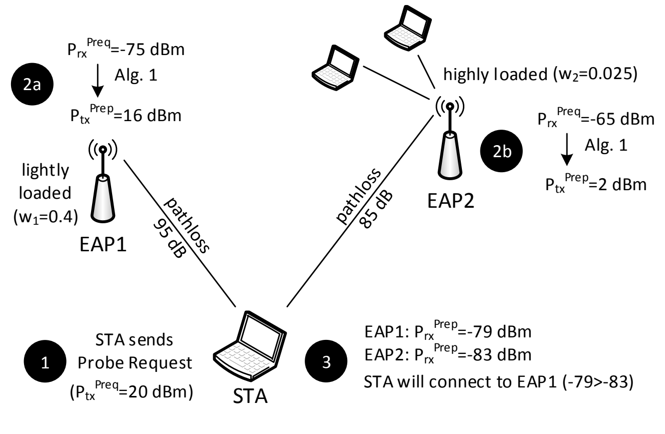

Fig. 6. Example illustrates the *Probe Response* TX power control in NxWLAN.

The complete pseudo-code for the calculation of the transmit power for the *PRES* frames is depicted in Algorithm 1. The algorithm is executed independently by each EAP upon the reception of a *PREQ*. Note, Algorithm 1 can be easily modified so that the visited EAP can enforce its own strategy (policy) as for the amount of resources (radio and backhaul) shared with the visiting clients. The helper function that maps the EAP's willingness to serve the client to the transmit power of the *PRES* frame is shown in Algorithm 2. Note, this algorithm takes the pathloss to the client into account to make sure that the *PRES* frame will be received with the correct receive power regardless of the distance between client and EAP (Fig. 5). Consider the following illustrative example (Fig. 6). Here the *PREQ* sent by the STA is received by two different EAPs at different received power levels $P_{\text{rx}}^{\text{Preq}}$, here -75 dBm and -85 dBm respectively. Depending on the network load, in radio and backhaul, each EAP computes its willingness to serve this client w, here 0.4 and 0.025 for the lightly and highly loaded EAP respectively. From both values $P_{\text{rx}}^{\text{Preq}}$ and w the TX power of the *PRES* $P_{\text{tx}}^{\text{Prep}}$ is estimated (see Fig. 5 with w and $P_{\text{rx}}^{\text{Preq}}$ on x and y-axis respectively.), here 16 dBm and 2 dBm respectively. The client receives two *PRES* packets and connects to EAP1 as the *PRES* is received with higher power $P_{\text{rx}}^{\text{Prep}}$.

The function *getMACRate()* calculates the maximum client MAC layer throughput. When considering a single WLAN BSS and assuming 802.11 DCF packet level fairness and no external interference the function can be defined as follows:

$$\textit{getMACRate}() = \gamma_k \times R_{\text{PHY}}^k \tag{1}$$

where R_{PHY}^k is the physical layer rate of client k and γ_k is its relative airtime:

$$\gamma_k = \frac{1/R_{\text{PHY}}^k}{\sum_{i \in C} 1/R_{\text{PHY}}^i} \tag{2}$$

where C is the set of active clients in WLAN BSS.

F. Normal Mode of Operation

Dealing with 802.11 Control Frames: Every 802.11 unicast frame transmission has to be acknowledged by the receiver. The 802.11 standard defines that an ACK frame has to be sent after SIFS which equals $10\,\mu$s. Tunneling a frame between WTP and VAP through the Internet backhaul would introduce a delay in order of at least (tens of) milliseconds, which is a few orders of magnitude higher and according to standard would make the transmitter always assume that the packet was lost and should be unnecessarily retransmitted, thus wasting valuable radio resources. For those reasons, tunneling of ACK frames is infeasible and all low level functions of 802.11 MAC, including acknowledgment and channel access (DCF), has to be realized by the WTP. In general, because of timing requirements, all control frames have to be generated by the wireless interface of the WTPs.

MAC Rate Adaptation: Wireless channels are extremely variable and their quality may change very fast due to many factors, like interference and fading. Therefore, the rate adaptation has to be done quickly to keep up with

Algorithm 1 Computes the TX power of *Probe Response* frames for RAP and all VAPs represented by their WTPs.

```
Require: $P_{\mathrm{rx}}^{\mathrm{Preq}}$    ▷ Receive power of Probe Request transmitted by scanning client station.
 1: procedure CALCPROBERESPONSETXPOWER
 2:     $\mathbb{C} \leftarrow \mathrm{GetFatClients}()$    ▷ Get the clients served by this EAP which are generating large network traffic.
 3:     $R^*_{\mathrm{PHY}} \leftarrow \mathrm{EstimatePhyRate}(P_{\mathrm{rx}}^{\mathrm{Preq}})$    ▷ Estimate the expected PHY rate of the new client when served by this EAP using the
    measured RX power from Probe Request (look-up table).
 4:     $\mathrm{rapDlBh} \leftarrow \mathrm{GetAvailableDlBackhaul}()$    ▷ Get available downlink backhaul capacity of EAP.
 5:     $\mathrm{rapUlBh} \leftarrow \mathrm{GetAvailableUlBackhaul}()$    ▷ Get available uplink backhaul capacity of EAP.
 6:     $R^*_{\mathrm{MAC}} \leftarrow \mathrm{GetMacRate}(\mathbb{C}, R^*_{\mathrm{PHY}})$    ▷ Predict expected client capacity at MAC layer when served directly (RAP).
 7:     $R^*_{\mathrm{ALL}} \leftarrow \min(R^*_{\mathrm{MAC}}, \mathrm{rapDlBh})$    ▷ Take minimum of available DL backhaul and wireless capacity.
 8:     $w^* \leftarrow \min(1, R^*_{\mathrm{ALL}}/R^{\mathrm{MAX}}_{\mathrm{ALL}})$    ▷ Calculate AP's willingness to serve client directly using RAP (normalized with max rate).
 9:     $P_{\mathrm{tx,rap}}^{\mathrm{Prep}} \leftarrow \mathrm{EncodeEapWillingness}(P_{\mathrm{rx}}^{\mathrm{Preq}}, w^*)$    ▷ Encode AP's willingness into TX power of Probe Response.
10:     for all $\mathrm{vap} \in \mathrm{GetVaps}()$ do    ▷ For each neighboring VAP represented by their WTP on this EAP.
11:         $\mathrm{vapDlBh} \leftarrow \mathrm{GetReportedDlBackhaul}(\mathrm{vap})$    ▷ Get reported available DL backhaul capacity of neighbor EAP.
12:         $\mathrm{vapUlBh} \leftarrow \mathrm{GetReportedUlBackhaul}(\mathrm{vap})$    ▷ Get reported available UL backhaul capacity of neighboring EAP.
13:         $R^{\mathrm{vap}}_{\mathrm{ALL}} \leftarrow \min(R^*_{\mathrm{MAC}}, \mathrm{rapDlBh})$    ▷ Take the minimum of available backhaul and wireless capacity.
14:         $\mathrm{R}'^{\mathrm{vap}}_{\mathrm{ALL}} \leftarrow \min(R^{\mathrm{vap}}_{\mathrm{ALL}}, \mathrm{vapDlBh}, \mathrm{vapUlBh})$    ▷ Consider packet tunneling from WTP to VAP.
15:         $w^{\mathrm{vap}} \leftarrow \min(1, \mathrm{R}'^{\mathrm{vap}}_{\mathrm{ALL}}/R^{\mathrm{MAX}}_{\mathrm{ALL}})$    ▷ Calculate EAP's willingness to serve client indirectly via WTP.
16:         $P_{\mathrm{tx,vap}}^{\mathrm{Prep}} \leftarrow \mathrm{EncodeEapWillingness}(P_{\mathrm{rx}}^{\mathrm{Preq}}, w^{\mathrm{vap}})$    ▷ Encode EAP's willingness into Probe Response TX power.
17:     end for
18:     return $P_{\mathrm{tx,rap}}^{\mathrm{Prep}}, P_{\mathrm{tx,vap_1}}^{\mathrm{Prep}}, \ldots, P_{\mathrm{tx,vap_N}}^{\mathrm{Prep}}$    ▷ Return the TX power values for all Probe replies (RAP and all VAPs represented by their
    WTPs).
19: end procedure
```

Algorithm 2 Helper function encodes EAP's willingness to serve the given STA into the transmit power of the *Probe Response* frame, i.e. high values means high willingness (e.g. low network load at EAP and/or better link quality).

```
Require: $P_{\mathrm{rx}}^{\mathrm{Preq}}$    ▷ Received power (dBm) from client's Probe Request.
Require: $w$    ▷ EAP's willingness to serve this client.
Require: $P_{\mathrm{tx}}$    ▷ The default STA transmit power (to be known).
Require: $P_{\mathrm{rx}}^{\mathrm{low}}$    ▷ The minimum receive power required for reception of Probe Request, e.g. -90 dBm
Require: $P_{\mathrm{rx}}^{\mathrm{high}}$    ▷ The max possible receive power of Probe Request, e.g. -50 dBm.
 1: procedure ENCODEEAPWILLINGNESS
 2:     $PL \leftarrow P_{\mathrm{tx}} - P_{\mathrm{rx}}^{\mathrm{Preq}}$    ▷ Pathloss of link AP-STA.
 3:     $P_{\mathrm{tx}}^{\mathrm{min}} \leftarrow \max(1, P_{\mathrm{rx}}^{\mathrm{low}} + PL)$    ▷ Minimum TX power to be used to avoid outage.
 4:     $P_{\mathrm{tx}}^{\mathrm{Prep}} \leftarrow \min(P_{\mathrm{tx}}, P_{\mathrm{tx}}^{\mathrm{min}} + w \times (P_{\mathrm{rx}}^{\mathrm{high}} - P_{\mathrm{rx}}^{\mathrm{low}}))$    ▷ Assumption: STA sends Probe Request with max power.
 5: return $P_{\mathrm{tx}}^{\mathrm{Prep}}$    ▷ Return Probe Response TX power (dBm).
 6: end procedure
```

wireless channel dynamics, i.e. in order of milliseconds. As in NxWLAN frames between WTPs and VAPs are tunneled over the Internet, an additional delay is introduced. Therefore, we argue that the WTP has to be responsible for the rate control.

III. NxWLAN's Implementation Details

Fig. 7 shows an overview of the NxWLAN implementation. We have implemented our prototype using open-source tools and we deployed it on small-form-factor-PCs based on Intel NUCs running Ubuntu 14.04. Due to space limitation, this section presents only the general overview of our prototypical implementation. A more detailed description can be found in our technical report [12].

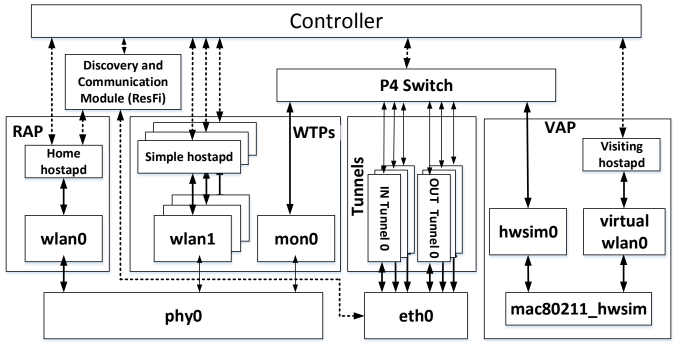

Fig. 7. NxWLAN implementation details - components inside a single residential Enhanced AP.

A. Discovery and Communication Module

We adapted the existing open-source implementation of ResFi [9] for over-the-air discovery and communication between neighboring EAPs. The ResFi [9] framework was originally designed to enable distributed Radio Resource Management (RRM) within residential WLAN deployments, e.g. distributed radio channel assignment [13]. Radio interfaces of participating APs are used for the efficient discovery of adjacent APs operating on the same or different radio channels, and to create a side-channel for the exchange of connection configuration parameters (like the public IP address of AP's RRM unit) and security credentials. Those data enable in turn the setting up of secured communication tunnels between adjacent APs via the (wired) Internet.

B. RAP, VAP and WTP

The RAP is deployed using a modified version of *hostapd* [14], that implements over-the-air discovery. The RAP is running on a virtual wireless interface, created on top of a physical wireless device.

The enabler for implementation of the VAP is the *hw_sim* [15] Linux kernel module. This module is usually used for testing of the 802.11 MAC subsystem functionality, as it allows the emulation of wireless transmissions on a single host machine. We modified this module to be able to sniff and inject 802.11 frames from and to the virtual wireless channel. The VAP contains a standard version of *hostapd*.

The WTP is responsible for: i) sniffing 802.11 frames by using a monitor interface which is created on top of a physical wireless device, and sending them to the switch, ii) receiving frames from the switch and injecting them to a monitor interface for over the air transmission. The frames are encapsulated with a RadioTap header. Moreover, the WTP contains a simplified version of *hostapd*, which is necessary for enabling ACK frame generation and rate control.

C. 802.11 Frames Tunneling

Tunneling of 802.11 frames between the VAP and the WTPs is realized using L2TP tunnels [16], which are created after the EAP neighbor discovery during the bootstrap phase. Basically, the L2TP protocol is used for interconnecting LAN networks over Internet.

The switching of the 802.11 frames between the tunnels and the monitor interfaces is performed by a Software Defined Networking (SDN) software switch. The switch was implemented in P4 [17], which is a new approach for SDN [18] providing a high-level language for programming the forwarding plane of packet processors. Our switch forwards native 802.11 frames based on their destination MAC address. The switch contains programmable match-tables that allows controlling its behaviour.

D. Probe Response TX Power Programming

We modified the ath9k wireless driver to allow the programming of the transmission power of *PRES* frames. Every time, the driver is handling a *PRES* frame transmission, it takes the appropriate entry from the TX power table filled before by the EAP controller via the debugFS interface. The appropriate TX power entries are calculated according to the algorithm presented in Sec. II-E.

E. MAC Rate Adaptation

For rate control, we use the Minstrel [19] algorithm as it is the most commonly used rate adaptation algorithm for 802.11 networks in Linux based systems. However, we faced a couple of problems, when trying to use it for traffic injected over a monitor interface, i.e. injected frames were not rate controlled and hence always transmitted at the basic rate. Therefore, we used the approach as suggested in [20], [21] which enables to add client STAs (and therefore entries in the rate table for them) to a wireless interface manually.

IV. Evaluation

This section is divided into three parts. First, we discuss the selection of the parameters for our evaluation scenarios. Second we demonstrate the feasibility of NxWLAN's implementation using COTS equipment and provide first results of its performance obtained in a small scale 802.11 testbed. Afterwards, we extend our performance studies by using network simulations in ns-3.

A. End-to-end parameter selection

Since end-to-end bandwidth and latency between two neighboring EAPs has significant impact on the operation and performance of NxWLAN, it is important to select those parameters properly for our test scenarios. Unfortunately, to the best of our knowledge, there is no large scale measurements of end-to-end connection parameters, especially between residential clients of different ISPs. Due to lack of reliable data, we were forced to deduce and estimate those parameters.

It is important to note that with NxWLAN we are targeting highly populated residential areas. Moreover, we assumed and considered the following characteristics. In most cities there are only a few different ISPs in each city, which makes it highly probable that some of the neighboring EAPs are connected to the same one. In such cases end-to-end delay between two EAPs should be low. However, in cases in which cable modems are used as access technology, subscribers (to some extent) are sharing their available bandwidth. Finally, we argue that in future, highly populated areas have the best chances to be repeatedly updated with the newest access technologies, like Fiber-to-the-Building/Home (FTTx).

1) Downlink and Uplink Speeds: A first approximation of user downlink and uplink speeds can be obtained from on-line tools like *www.speedtest.net*. Such tools allow users to test their Internet connection capacity and collect those measurements. In advance, statistics of average connection speeds for countries and even for some bigger cities are provided. For example, in Berlin, the average connection performance is 36 Mbps in downlink and almost 9 Mbps in the uplink, while the performance of the top 10% is 94.5 Mbps and 11.9 Mbps, respectively. In New York, it is 68/27 Mbps and 154/66 Mbps, respectively. Of course, presented measurements show the link parameters between user and the closest test server. Nevertheless, we argue they are sufficient for first approximation of links between two neighboring EAPs.

2) End-to-End Delay: In order to get first estimation of the end-to-end delay between residential clients, we have performed small scale experiments in Berlin. We measured the RTT using the *ping* tool between three residential clients connected to different ISPs in different parts of the city. Our results have shown that average RTT is in range of 40-50 ms.

The end-to-end delay between two residential clients A and B is composed of: access delay of client A; delay between border router of A's ISP and Internet Exchange Point (IXP); delay between IXP and border router of B's ISP; and access delay of B. If both users are subscribed to the same ISP the end-to-end delay consist only of access delays. In [22] authors measured and reported that the access delay for most of residential users is in the range of 0–10 ms. This is consistent with results from Japan performed in 2009 and presented in [23]. As reported, the access delay for four biggest ISPs in big cities in Japan was around 1-2 ms. Also, data presented in [24] confirms range of 0-10 ms. Hence, we can assume that access delay equals 10 ms in worst case scenario.

Unfortunately, we did not find similar measurement reports for delay between ISPs and IXPs. However, Feldmann *et al* [25] found out that in many cases for residential Internet connections the local (access) RTT dominates the remote (end-to-end) RTT, i.e., it takes longer to get to the Internet than traveling the Internet. They also reported that local RTT is on average 7 ms, while a remote RTT of 13 ms was reported for the users and servers located in Europe. As we are targeting highly populated areas, we can assume that there is at least one IXP in each bigger city, which minimizes the delay between two ISPs. Taking all of this into account, we assume that delay between ISP and IXP in no higher than 2 ms.

Considering all our assumptions, the estimated end-to-end delay equals 24 ms and RTT 48 ms, what is consistent with our small scale measurements (40-50 ms).

Finally, the maximal RTT is bounded by the time the STA stays on single channel waiting for *PRES* after sending *PREQ*. In [10], the scanning interval was reported to be 100 ms (or higher) for most Wi-Fi cards. If the RTT between two EAPs is higher than this interval, NxWLAN simply does not work. For that reason, EAPs should measure RTT before setting up NxWLAN relationship, and abandon such try in case RTT exceeds predefined scanning interval.

3) Conclusions: Having investigated available measurement data, we argue that 50 Mbps of bandwidth and an end-to-end delay of 50 ms between two residential clients are feasible. In our experiments, we always use RTT of 50 ms (as the worst case scenario), while for Internet connection, we evaluated several different combinations of downlink and uplink speeds.

B. Testbed Results

1) Methodology: The proposed NxWLAN approach is analyzed by means of experiments in a small indoor 802.11n/a testbed. The setup shown in Fig. 8 mimics two adjacent residential apartments each equipped with a single EAP. During the experiment a client station was placed at ten different locations within Bob's apartment as shown in the figure.

The hardware used for the EAPs and client stations were standard x86 machines with *Ubuntu 14.04* and *Atheros* Wi-Fi NICs using AR9280 chipsets. For our experiments we set the two EAPs on two different not otherwise used channels of the 5 GHz ISM band, i.e. channel 40 and 44. The physical layer was set to 802.11a. We used the traffic control tool [26] to emulate latency - 50 ms - and backhaul capacity -UL and DL of 50 Mbit/s - between two EAPs.

We considered the following two scenarios: *i)* without any background traffic and *ii)* with high background traffic on home EAP and lightly loaded neighboring EAP. In both cases we measured the downlink TCP/IP throughput towards the client station using the iperf tool [27]. We compare NxWLAN with a baseline in which the client station is always served by the home EAP.

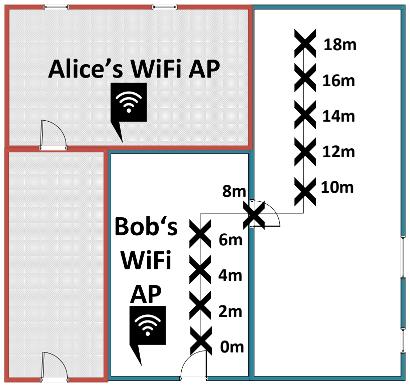

Fig. 8. Experiment setup mimics two adjacent residential apartments each with a single EAP.

2) Results: **Experiment 1: (Extended coverage)** The objective of this experiment is two-fold. First, we want to show that the proposed approach is, indeed, able to increase the coverage of the home WLAN by using the neighboring EAP. Second, that the visiting EAP is able to provide a high throughput to the client station although the 802.11 data traffic is tunneled to the home EAP over the emulated Internet connection. This shows the feasibility of the approach, as well as an example of the gain which this solution can provide.

Results 1: Fig. 9 shows the mean and standard error of the downlink throughput to the client station at the ten different locations. Every location point was measured 10 times over a period of 10s. The results reveal that in contrast to baseline NxWLAN is able to provide coverage even at the far corner of the apartment. Moreover, the client achieves a high throughput (close to the maximum), while being served by the WTP on the neighboring EAP (locations 10-18 m).

Experiment 2: (Load balancing) In this experiment we consider a scenario with unequal network load, i.e. the home EAP (Bob) is highly loaded. As example for such load we use two client stations with bulk TCP data transfers, whereas the neighboring EAP (Alice) remains unused (idle). We demonstrate that NxWLAN is able to steer the client device to the lightly loaded EAP for load balancing reasons by means of manipulating the transmit power of the *PRES* frame.

Results 2: From Fig. 10 we observe that compared to baseline, NxWLAN is able to dramatically increase the downlink throughput of the client. This is because in baseline the client station has to share the wireless medium with the two backlogged low-bitrate users (PHY rate of 6 Mbps) in the home EAP whereas in NxWLAN the client is steered to the neighboring idle EAP (Alice) from the very beginning. Even at very close locations to the home EAP our proposed transmit power control for the *PRES* frames is able to successfully steer the client device to the far, but lightly loaded neighboring EAP.

Note that in the baseline case, the performance of the STA using rate control algorithm suffers due to packet fairness, i.e. it gets access to the channel with the same probability as the STAs with the fixed slow rate (6 Mbps), thus most of the time it waits for access and its channel occupancy time is the shortest one. This issue can be solved by using Transmit Opportunity, here 1.5 ms.

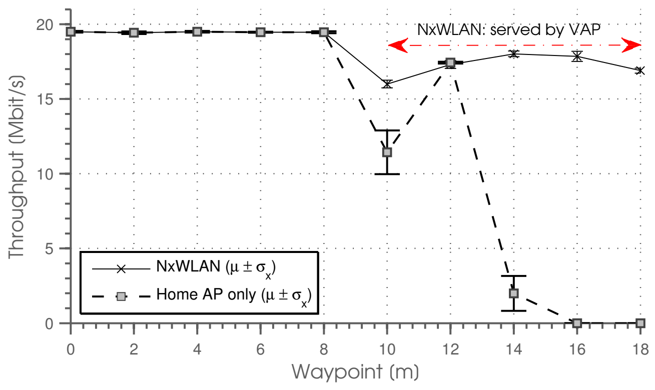

Fig. 9. No background traffic: with NxWLAN the client STA is served by WTP in Alice's EAP after 12 m way-point.

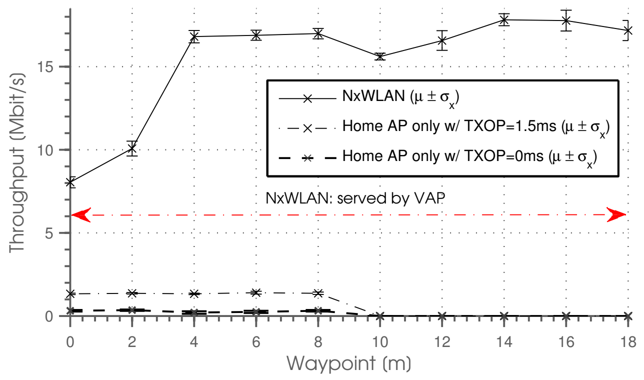

Fig. 10. Congested home EAP (Bob): with NxWLAN the client STA is always served by WTP in Alice's EAP.

C. Simulation Results

1) Methodology: We analyzed the performance of larger configurations of NxWLAN by means of network simulations using ns-3 [28]. We have considered a scenario with five co-located EAPs in an area of the size of $300\,m \times 300\,m$, i.e. one Home EAP in the middle and four neighboring EAPs at a distance of 75 m to the Home EAP operating on different channels.

For EAP's Internet connection, we evaluated couple different combinations of DL and UL speeds and set the round trip time delay to 50 ms.

For the wireless access, we used the 802.11n standard with HT40, short Guard Interval and Minstrel rate adaptation algorithm. All EAPs operated on different RF channels in 5GHz ISM band. During a single simulation, the client STA was placed in one of the locations in the grid, calculated as $(30x, 30y), x, y \in \{0, 1, \ldots, 10\}$; a single DL TCP bulk flow from a server in the backhaul to the STA was set-up and its throughput was measured. For each position the simulation was repeated 25 times, simulation time was set to 10 s and results from first 3 s were discarded to exclude the warm-up period.

We conducted two kinds of experiments and for both we ran a campaign of simulations without (baseline) and with NxWLAN enabled.

2) Results: **Experiment 1: (Extended coverage)** In this experiment the objective is to show that coverage of the home WLAN can be extended by utilizing the neighboring EAPs. As already mentioned, each EAP is operating on a different RF channel and there is only one STA receiving data. Thus, there are no interferences between EAPs and the extension and throughput gains are achieved with NxWLAN only by reduction of pathloss, i.e. STA always connects to closest EAP with strongest signal.

Results 1: As shown in Fig. 11, NxWLAN offers a large gain in case the backhaul UL speed is high, e.g. when DL/UP equals 100/50 Mbps it offers on average a gain of STA throughput of 2.5×. Note, that even for a very slow backhaul configuration, like 20/5 Mbps, NxWLAN still offers on average a gain comparing to baseline, i.e. 1.24×, which is due to improving the performance at the cell edge of the Home EAP.

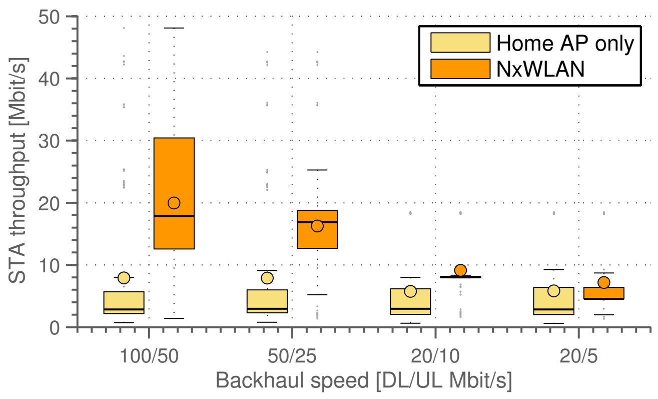

Fig. 11. No background traffic.

Experiment 2: (Load balancing) To evaluate the performance of NxWLAN with unequally loaded EAPs, we created a similar setup as in Experiment 1. In addition to the client STA of interest, we added also three additional client STAs, that were always connected to the Home EAP and located at distance of 0, 20 and 40 m from it. They were responsible for generation of background traffic and saturating the Home EAP cell. To this end, each of them was an endpoint of single TCP bulk flow from a server in backhaul.

Results 2: From Fig. 12 we can observe that with high speed backhaul, NxWLAN allows the STA to achieve a high throughput gain as compared to baseline, e.g. with 200/100 Mbps backhaul NxWLAN offers on average a gain of 7.7×. In case of a slower backhaul, the backhaul itself becomes the bottleneck and therefore the gain of NxWLAN is limited, e.g. with 50/25 Mbps backhaul, NxWLAN provides on average gain of $1.9\times$.

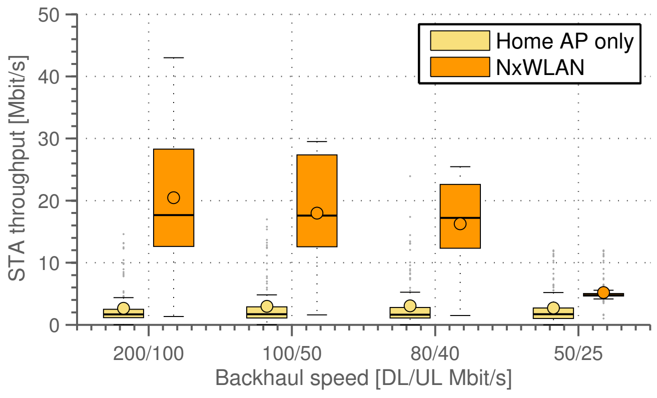

Fig. 12. Congested home EAP.

D. Discussion

Results from our experiments and simulations are consistent and prove that NxWLAN is able to provide significant throughput gains in residential environments. The gain is achieved due to NxWLAN's ability to steer new client stations towards the EAP that provides the highest available capacity by taking both the radio and the backhaul network into account. In this way radio and backhaul resources are used most efficiently.

Results from larger topologies and configurations were left out due to space limitation.

V. Related Work

Related work falls into two categories:

WLAN sharing for Internet access: In WiseFi [3] the authors presented a concept featuring a centralized server managing the reciprocal Wi-Fi sharing between APs located within a neighborhood. Such approach assumes full centralization, including the centralized storage and management of access credentials, features which we avoid in NxWLAN.

Efstathiou et al. [29] envision a city-wide Wi-Fi access for all participating members (every participant is obliged to provide free Wi-Fi access to other participants). Participants create their individual identities (public-private key pairs) and receive signed digital receipts when they provide Wi-Fi service to other participants. On the basis of usage accounting real cooperation and suppression of free-riders is enabled. NxWLAN is different as it does not require either new types of identity beyond the one used when connected to the own AP, nor any additional configuration/application on the client stations.

A similar system for Wi-Fi sharing communities was proposed, called PISA, it was presented in [30]. Despite having similar general concepts (i.e. tunneling and using user's home infrastructure for Internet access), it differs from NxWLAN in three main points: *i)* it tunnels IP packets between STA and home AP, which requires additional software to be installed in the client device; *ii)* it requires users to register their home AP and mobile devices, so their membership can be verified while accessing the visiting APs; and *iii)* in PISA a visiting AP is responsible for all 802.11 control and management functions.

Recently the idea of sharing Wi-Fi Access is followed in the so-called community Wi-Fi networks in which residential Wi-Fi AP owners are sharing some part of their capacity with members of the community, e.g. Guifi.net [31]. Lastly, also commercial Internet service providers start utilizing such approaches by piggybacking their hot-spots on residential Wi-Fi APs, e.g. [32], [33]. However, usually usage of these solutions is limited only to clients of a single ISP, i.e. only clients of an ISP are able to connect to its *virtual* hot-spots. In contrast, using NxWLAN neighbors have a possibility of sharing their APs even being clients of different ISPs.

Windows 10 introduced Wi-Fi Sense[34] that allows the sharing of the security keys of a Wi-Fi network to which an user was connected, with her contacts listed in Outlook, Skype, Hotmail and optionally also Facebook. The feature was controversial from privacy and security point of view. A lot of users got suspicious and disabled it. Finally, in May 2016 Microsoft removed it completely. Contrary, NxWLAN provides end-to-end security with individual passwords not shared with any additional instances.

AP virtualization: The idea of removing the management functions of the MAC from individual APs and shifting them to a dedicated instance has been introduced CAPWAP [35] and LWAPP [36] with the clear goal to enable centralized management of multiple APs by the wireless access service provider. Both of these solutions require state sharing between the AP involved and the centralized management instance – while the later one is storing all access credentials. Both features are avoided in NxWLAN.

In the CloudMAC approach, Vestin et al. [37] have shifted the processing of MAC frames in enterprise WLAN systems from the APs to the cloud with the dual goal of i) allowing a simplistic – and thus cheap – AP solution, ii) supporting centralized management. NxWLAN follows the idea to offload the processing of management MAC frames, but in contrast to CloudMAC, we follow different goals and keep a single location of processing for each user – her home router – featuring a fully distributed solution.

Anyfi [38] is a commercial product, which enables a similar user experience as NxWLAN. Likewise, an unmodified client device uses the WPA pass-phrase already stored in the device to authenticate against its home AP. However, Anyfi differs in the following points: i) it puts the focus in providing seamless worldwide roaming whereas NxWLAN aims to improve the coverage/capacity at home WLAN by using neighboring APs, ii) it requires the home AP to register its IP and MAC addresses of all of its clients in a cloud controller – the mapping is used by other APs during the discovery procedure when the client is roaming. Note that, if MAC-randomization is used, the visited AP is not able to discover and setup a connection to the client‘s home AP. In contrast, NxWLAN, thanks to the limited number of neighboring APs, performs discovery using a broadcast mechanism, which assures its operation even in case of MAC-randomization.

VI. Conclusions & Future Work

This paper introduced NxWLAN, the first system that enables a secure extension of user's home WLAN through the usage of neighboring APs in residential environments without revealing encryption keys to potentially untrusted neighbor APs. We have implemented and evaluated a NxWLAN prototype using real COTS hardware in a mimicked residential scenario as well as through simulations. The source code is provided to the community as open source. Our results showed that NxWLAN is able to increase the radio coverage of the home WLAN and improve users' performance, while taking the load of the wireless radio channel and backhaul network of both the neighboring and the home APs into account.

For future work, we plan to provide mobility support using the handover mechanisms used in enterprise networks [20] and apply MAC layer slicing [39] to provide airt-time guarantee mechanisms and traffic isolation between the WTP and the RAP. Furthermore, a possible extension is to support an additional backhaul bundling together with multi-path TCP (MPTCP) that can be used in situations in which a STA is served by multiple APs simultaneously as proposed by Bayer et al. [40].

VII. Acknowledgment

This work has been supported by the SDN WiFi Community project funded by Deutsche Telekom and European Union's H2020 research and innovation programme (GA No 645274).

References

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper."

[2] D. Watkins, "Global Broadband and WLAN (Wi-Fi) Networked Households Forecast 2010-2019."

[3] J. Shi, L. Gui, D. Koutsonikolas, C. Qiao, and G. Challen, "A little sharing goes a long way: The case for reciprocal WiFi sharing," in *ACM HotWireless*, 2015.

[4] L. Oliveira, K. Obraczka, and A. Rodríguez, "Characterizing User Activity in WiFi Networks: University Campus and Urban Area Case Studies," in *ACM MSWiM*, 2016.

[5] M. Balazinska and P. Castro, "Characterizing mobility and network usage in a corporate wireless local-area network," in *MobiSys*, 2003.

[6] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "Softran: Software defined radio access network," in *ACM SIGCOMM*, 2013.

[7] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo, "Large-scale Measurements of Wireless Network Behavior," in *ACM SIGCOMM*, 2015.

[8] Wikipedia, "German law: Störerhaftung," *https://de.wikipedia.org/wiki-/St%C3%B6rerhaftung*, 2016, accessed: 2017-04-20.

[9] S. Zehl, A. Zubow, M. Döring, and A. Wolisz, "ResFi: A Secure Framework for Self Organized Radio Resource Management in Residential WiFi Networks," in *IEEE WoWMoM*, 2016.

[10] G. Castignani, A. Arcia, and N. Montavont, "A Study of the Discovery Process in 802.11 Networks," *ACM SIGMOBILE*, 2011.

[11] Android.com, "Android 6.0 changes," 2016, [Online; accessed 23-March-2017]. [Online]. Available: https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html

[12] P. Gawłowicz, S. Zehl, A. Zubow, and A. Wolisz, "NxWLAN: Neighborhood eXtensible WLAN," 2016.

[13] S. Zehl, A. Zubow, and A. Wolisz, "Practical distributed channel assignment in home Wi-Fi networks," in *IEEE 18th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Macau, P.R. China, Jun. 2017.

[14] J. Malinen, "hostapd," *https://w1.fi/hostapd/*, accessed: 2017-04-20.

[15] Linux Community, "MAC 802.11 HwSim," *https://wireless.wiki.kernel.org/en/users/drivers/mac80211_hwsim*, accessed: 2017-04-20.

[16] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer Two Tunneling Protocol L2TP," Tech. Rep., 1999.

[17] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM*, 2014.

[18] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM*, 2008.

[19] D. Xia, J. Hart, and Q. Fu, "Evaluation of the Minstrel rate adaptation algorithm in IEEE 802.11g WLANs," in *IEEE ICC*, 2013.

[20] A. Zubow, S. Zehl, and A. Wolisz, "BIGAP: Seamless handover in high performance enterprise IEEE 802.11 networks," in *IEEE NOMS*, 2016.

[21] S. Zehl, A. Zubow, and A. Wolisz, "BIGAP – A seamless handover scheme for high performance enterprise IEEE 802.11 networks," in *IEEE NOMS*, 2016.

[22] S. Sundaresan, W. De Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè, "Broadband Internet performance: a view from the gateway," in *ACM SIGCOMM*, 2011.

[23] K. Yoshida, Y. Kikuchi, M. Yamamoto, Y. Fujii, K. Nagami, I. Nakagawa, and H. Esaki, "Inferring pop-level isp topology through end-to-end delay measurement." 2009.

[24] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing Residential Broadband Networks," in *ACM SIGCOMM*, 2007.

[25] G. Maier, A. Feldmann *et al.*, "On dominant characteristics of residential broadband internet traffic," in *ACM SIGCOMM*, 2009.

[26] B. Hubert, "TC - Manpage," *http://lartc.org/manpages/tc.txt*, 2001, accessed: 2017-04-20.

[27] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "iperf," *https://iperf.fr/*, accessed: 2017-04-20.

[28] "ns-3 network simulator," *www.nsnam.org*, accessed: 2017-04-20.

[29] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos, "Controlled Wi-Fi sharing in cities: A decentralized approach relying on indirect reciprocity," *Mobile Computing, IEEE Transactions on*, 2010.

[30] T. Heer, S. Gotz, E. Weingartner, and K. Wehrle, "Secure Wi-Fi Sharing at Global Scales," in *IEEE ICT*, 2008.

[31] Guifi.net, "Guifi.net - Commons Telecommunications Network Open, Free and Neutral," *http://guifi.net/*, 2016, accessed: 2017-04-20.

[32] H. Kischkewitz, "IP-Serie: Mit WLAN TO GO freies WLAN weltweit nutzen," *http://blog.telekom.com/2016/02/26/wlan-to-go/*, 2016, accessed: 2017-04-20.

[33] Fon-Network, "Fon is the Global WiFi Network," *https://fon.com/*, 2016, accessed: 2017-04-20.

[34] B. Krebs, "Microsoft Disables Wi-Fi Sense on Windows 10," *http://krebsonsecurity.com/2016/05/microsoft-disables-wi-fi-sense-on-windows-10/*, 2016, accessed: 2017-04-20.

[35] S. Govindan, H. Cheng, Z. Yao, W. Zhou, and L. Yang, "Objectives for control and provisioning of wireless access points (CAPWAP)," Tech. Rep., 2006.

[36] P. Calhoun, "Lightweight Access Point Protocol," *RFC 5412*, 2010.

[37] J. Vestin, P. Dely, A. Kassler, N. Bayer *et al.*, "CloudMAC: towards software defined WLANs," *ACM SIGMOBILE*, 2013.

[38] Anyfi-Networks, "Anyfi Networks SIMPLE," *http://www.anyfinetworks.com/*, 2016, accessed: 2017-04-20.

[39] S. Zehl, A. Zubow, and A. Wolisz, "Hotspot slicer: Slicing virtualized home Wi-Fi networks for Air-Time guarantee and traffic isolation," in *IEEE 18th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Macau, P.R. China, Jun. 2017.

[40] N. Bayer, A. Girmazion, M. Amend, K. Haensge, R. Szczepanski, and M. Hailemichael, "Bundling of DSL resources in home environments," in *IEEE WoWMoM*, 2016.