TKN Telecommunication Networks Group

Technische Universität Berlin

Telecommunication Networks Group

BIGAP – Seamless Handover in High Performance Enterprise IEEE 802.11 Networks

Anatolij Zubow, Sven Zehl and Adam Wolisz

{zubow, zehl, wolisz}@tkn.tu-berlin.de

Berlin, September 2015

TKN Technical Report TKN-15-0004

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

Enterprise IEEE 802.11 WiFi networks need to provide high network performance to support a large number of diverse clients like laptops, smartphones and tablet as well as capacity hungry and delay sensitive novel applications like mobile HD video & cloud storage. Moreover, such devices and applications require much better mobility support and higher QoS/QoE. Existing solutions can either provide high network performance or seamless mobility but not both.

We present BIGAP, a novel architecture providing both high network performance as well as seamless handover. The former is achieved by assigning different channels to co-located APs to fully utilize the available radio spectrum. The latter is achieved by providing a mechanism for below MAC-layer handover through exploiting the Dynamic Frequency Selection capability in IEEE 802.11.

The proposed solution is fully compatible with 802.11 and requires no modifications to the wireless clients.

We present testbed results showing a significant improvement in terms of network outage duration, i.e. $32\times$ smaller as compared to state-of-the-art 802.11 handover scheme, and negligible throughput degradation during handover operation. This allows the realization of mobility and load balancing schemes requiring frequent and seamless handover operations.

Contents

Chapter 1

Introduction

There is a clear trend towards deploying IEEE 802.11 wireless networks (WiFi) in enterprise environments as a replacement for wired networks. According to the Cisco Visual Networking Index, by 2019, 53 percent of monthly IP traffic will be delivered over WiFi and mobile devices [8]. Moreover, the way enterprise WiFis are used changed dramatically. WiFi allows enterprise customers to enjoy mobility indoor and outdoor. The appearance of WiFi enabled laptops, Bring Your Own Devices (BYOD) like smartphones and tablets require much better mobility support and higher QoS/QoE.

However, the deployment of 802.11 enterprise networks is challenging. Providing just coverage at all locations is not sufficient anymore. In addition a high network capacity is required to fulfill properties like high throughput, low latency, high reliability and QoS to be able to support capacity hungry novel applications like multimedia streaming applications, mobile HD video, social networking & cloud storage.

Enterprise IT departments tackle this issue by a very dense deployment of Access Points (AP), i.e. an AP in each office room, to allow each client Station (STA) to connect with a very close AP. To avoid co-channel interference and competition between co-located APs which may become very severe in dense AP deployments, neighboring APs are operated on different RF channels. This is a promising approach as with the new 802.11ac standard the available spectrum in the 5 GHz further increases and is sufficient to allow channel reuse and segmentation of APs into separate collision domains even in dense AP deployments [5]. In particular there up to 25 non-overlapping channels available [1].

Although, mobile STAs in a dense WiFi network can choose from many possible APs, this degree of freedom is not fully exploited in 802.11 resulting in restricted mobility. This is because in standard 802.11 STAs select the APs they would like to associate using pure local information, e.g. signal strength. This is suboptimal especially if we consider non homogeneous scenarios where hotspot cells can have large number of STAs. Moreover, normally once associated STAs stay connected to the AP even if there is an AP which is able to provide better service quality, e.g. higher link quality or lower utilization [20]. However, slower STAs, i.e. those with a low quality link, use significantly more airtime to transmit the same amount of data which results in slower clients monopolizing airtime and hence significantly decreasing network capacity. Therefore, an infrastructure-initiated handover scheme which allows seamless mobility and client load balancing is of fundamental importance in Enterprise WiFi networks.

Unfortunately, operating APs on different RF channels complicates handover operations as the STAs have to switch their channel. Neither standard 802.11 nor proposals from the literature are able to provide seamless and efficient handover operation in multi-channel WiFi networks which is

[1] To be precise: there are 9 and 25 non-overlapping 20 MHz channels available respectivily depending on whether Dynamic Frequency Selection is supported or not.

however of integral importance in order to provide uninterruptible network connections and high QoS (e.g. for VoIP).

In this paper we present BIGAP, an architecture for enterprise WiFi networks, which is efficient, i.e. scales with the number of serving STAs and AP density, while providing support for seamless mobility management and load balancing. While approaches like Virtual Access Point (VAP [10]) and mVAP [1] are able to provide seamless mobility they do not scale with the number of STAs and AP density due to the large wireless signaling overhead required for managing the VAPs. Other approaches like DenseAP [16] provide high scalability due to use of advanced frequency planning but do not provide seamless mobility. BIGAP aims for practical applicability. In particular it does not require any hardware/driver changes on the client and AP side and is therefore fully compatible with commodity 802.11n/ac cards which support Dynamic Frequency Selection. BIGAP decides on the channel assignment to APs on a long-term basis whereas the decision by which AP a particular STA is served is based on short-term information like channel-state information (mobility) and traffic conditions (load balancing).

Contributions: The proposed architecture supports seamless handover of standard 802.11 compliant clients in high density (with respect to APs) multi-channel enterprise WiFi networks. The presented framework allows the implementation of novel mobility and load-balancing schemes which require frequent and seamless handover operations. The performance of proposed system is evaluated in a real testbed and compared to standard 802.11 and solutions from literature.

Chapter 2

IEEE 802.11 Primer

This section gives a brief overview of the relevant parts of the 802.11 standard.

2.1 BSSID and SSID

According to the 802.11 standard, a basic service set (BSS) is a set of stations that are logically associated with each other. Every BSS is identified by a BSSID, which is a 48 bit identifier used by all stations in a BSS within the frame headers. If an 802.11 network operates in the infrastructure mode, a BSS usually comprises a single AP and the STAs associated with it. If multiple APs are connected to a distribution system, the complete system with all single BSSes interconnected is called an extended service set (ESS). All access points in an ESS are using the same service set identifier (SSID) which serves as the network name visible to stations.

2.2 Automatic Repeat reQuest

The original IEEE 802.11 MAC conception required that each frame sent to the receiver has to be acknowledged separately. With the IEEE 802.11e extension, the block acknowledgment (BA) was introduced, which greatly improves the MAC efficiency. BAs enable the sender to send a stream of frames which can be acknowledged by the receiver using a single BA frame containing a bitmap of the received frames. A BA session starts with a setup phase and ends with a tear-down phase. In the setup phase, capability information such as buffer size and BA policy are negotiated. Afterwards, the transmitter can send frames without waiting for ACK frames. After the data is transferred, the sender requests the BA using a BA request frame (BAR). The BA session can be torn down either by the sender or the receiver by sending a delete BA (DELBA) frame.

2.3 Handover in Standard 802.11

Roaming between APs in IEEE 802.11 is entirely driven by STA decisions but the standard does not dictate how a STA makes its decision on how to switch between APs. Most STA devices use signal strength as the primary metric and start scanning for new APs when the signal strength to the currently associated AP is low. STA stickyness [20] is often a problem because most STA implementations try to stay connected to an AP as long as possible even if the signal quality is poor. After a STA decided to perform an handover, the following steps are taken: i) discovery (scanning), ii) (re)authentication and iii) (re)association. Further, if security is applied additional steps may be required. Apart from that, there is an additional delay caused by the time taken to update the ARP cache or routing changes

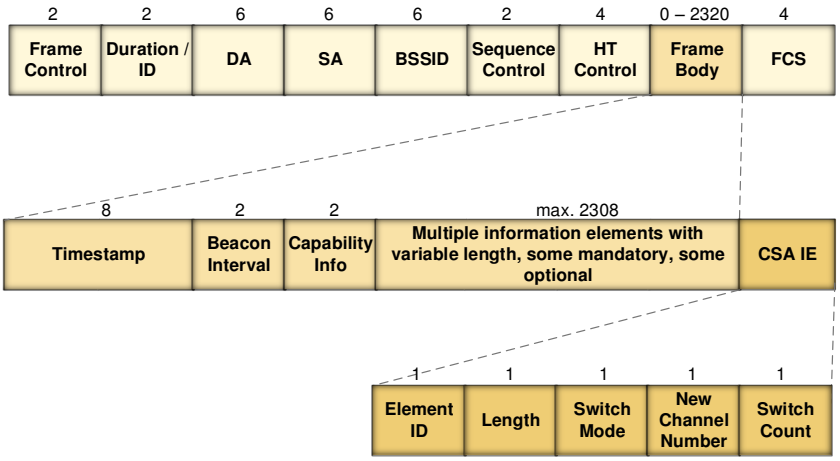

Figure 2.1: Structure of an IEEE 802.11 beacon frame with embedded CSA-IE.

in the wired backbone. Consequently, due to all these steps, there is always a significant network outage during which the STA is unable to send or receive data traffic.

2.4 Channel Switch Announcement

The majority of RF channels in the 5 GHz band require a mechanism termed dynamic frequency selection (DFS) [1]. The usage of DFS makes sure that channels used by radars are not used by APs and STAs. With DFS an 802.11 device continuously scans the current channel for other signals like radars, and switches to another channel if the current channel is occupied.

In 802.11 infrastructure mode an AP can inform its associated stations about the detection of a radar signal by transmitting a beacon frame with a Channel Switch Announcement Information Element (CSA-IE) together with the number of the new channel (Fig. 2.1). This functionality allows the AP and the associated stations to perform a coordinated channel switch, i.e. after the channel switch the stations remain associated with the AP. There is also the option to transmit the CSA IE in other 802.11 management frames.

[1] In Europe for 802.11 devices it is the spectrum in 5.25–5.35GHz and 5.47–5.725GHz.

Chapter 3

BIGAP's Design Principles

Enterprise WiFi architectures need to be optimized to support a large number of STAs while providing high QoS, throughput, reliability and low latency. To fully utilize the available radio spectrum at each location a high density deployment of APs is required [16]. However, enterprise WiFi networks also require strong support for mobility management and load balancing. Hence, there is a need for an STA handover scheme which can be controlled by the infrastructure.

Currently the only applicable approach for infrastructure-initiated handover which does not require modifications on the client devices is the DenseAP hard-handover scheme proposed by Murty et. al [16]. DenseAP's hard handover scheme removes the STA stickiness by transferring the handover decision from the client to the infrastructure, but leaves the outage duration caused by the amount of time the STA needs for the connection build-up with the new AP. This duration includes the delays caused by scanning/probing, authentication and reassociation.

BIGAP decreases the outage period and removes all aforementioned delays by transferring the current state of the STA from the serving AP to the target AP. To enable this possibility, the BIGAP topology uses a single global BSSID for the whole ESS and thereby for all APs. From the STAs point of view, the whole ESS including all APs seems like one BSS or one big AP. As the same BSSID operated on the same RF channel would cause collisions, duplicated frames in the backbone and would lead to a high channel utilization, BIGAP uses different RF channels for all co-located APs. For performing the handover process, BIGAP exploits the 802.11 DFS functionality and leads the STA to believe that the serving AP will perform a RF channel switch. In actual fact, the serving AP remains on its current RF channel but the target AP is operating on the new RF channel. Due to the fact that all APs use the same BSSID and due to the fact that the current state of the STA on the old AP was transferred to the new AP, the STA believes the new AP is the old AP which has also switched the RF channel. By relying on these principles the communication can be continued without any further outage except the time needed for RF channel switching.

BIGAP does not require any modifications to the client STAs but requires the support of 802.11n/ac which includes the IEEE 802.11h amendment. Further, BIGAP requires the existence of a sufficient large number of available RF channels so that different channels can be assigned to co-located APs[1].

BIGAP's general framework consists of two parts, cf. Fig 3.1. One component resides at the APs where it collects wireless statistics and executes BIGAP controller commands. The other component is the BIGAP controller which has a global view of the network state and allows the coordination of the handover operations between the serving and target AP. The BIGAP controller decides on the handover operation based on a policy which uses wireless statistics like average link quality and traffic conditions.

[1] Note, in the 5 Ghz band of 802.11 there are 9 and 25 non-overlapping 20 MHz channels respectively.

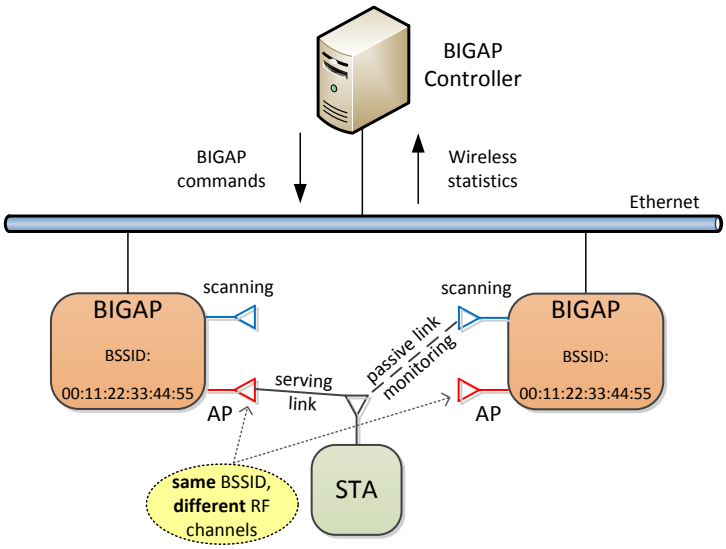

Figure 3.1: **The BigAP architecture.** The BIGAP controller coordinates the APs using the BIGAP interface to provide seamless handover. All BIGAP APs of the same network have the same BSSID. Co-located APs are operated on different channels.

In BIGAP each AP is equipped with two WiFi interfaces. The first one is operating in AP mode for serving client network traffic whereas the second interface is used for collecting wireless statistics like the quality of the wireless links of (not associated) STAs in communication range which is required to provide seamless mobility. In particular, this interface is operated in promiscuous monitor mode to collect information about overheard packets by periodically hopping over all channels used by neighboring BIGAP APs.

BIGAP exploits the possibility of DFS to announce channel switches to trigger a channel switch within STAs and further to perform the handover operation. To achieve this the BIGAP controller instructs the serving AP to send an beacon frame containing a CSA-IE with the RF channel of the target AP, cf. Fig. 2.1. Receiving this CSA-IE triggers the channel switching in the STAs to the desired RF channel. Since IEEE 802.11 beacon frames are layer 2 broadcast frames, this operation would trigger channel switching of all STAs associated with the serving AP or operating on the same channel in communication range [13]. BIGAP solves this problem by sending an unicast beacon frame destined to the selected STA, i.e. the 802.11 destination address is no longer broadcast but the unicast address of the particular STA. The selected channel determines implicitly the target AP since there is at most one AP using the same channel in a BIGAP collision domain.

Chapter 4

BIGAP - Detailed Specification

In this section we give a detailed description of the BIGAP specification.

4.1 AP Channel Assignment

When an BIGAP AP is turned on for the first time it performs the following steps. First, it starts scanning the whole 802.11 radio spectrum for neighboring BIGAP APs beacons. Therefore, it uses its scanning interface and reports for each detected AP the SSID, BSSID and the used channel. Second, it registers itself with the BIGAP controller. Third, the BIGAP controller retrieves the scanning report from the AP.

The controller configures the AP to use the same common BSSID in the BIGAP SSID network. Moreover, the controller decides on the channel to be used by this AP. Because in BIGAP all APs in the same SSID use the same BSSID the channel must be selected to guarantee collision-free channel assignment. That means that co-located BIGAP APs must operate on different channels to avoid MAC acknowledgment collision for uplink traffic. Hence, from the scanning results of the APs the controller constructs a network graph $G = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of BIGAP APs and there is an edge $e \in \mathscr{E}$ between two APs if they are in communication range. BIGAP uses the following heuristic according to the channels are assigned to APs in such a way that any channel is used at most once in its two-hop neighborhood [19]:

$$\forall v \in \mathscr{V} : \mathrm{ch}(v) \neq \mathrm{ch}(w), w \in \mathrm{nb}(\mathrm{nb}(v)) \tag{4.1}$$

here $\mathrm{nb}(x)$ represents direct neighbors of x while $\mathrm{ch}(x)$ represents the channel used by AP x.

4.2 STA Association

BIGAP supports both active and passive scanning. In BIGAP the AP for the initial STA association is selected by the STA itself. If the selected AP is not optimal with respect to the some algorithm running in the BIGAP controller (e.g. load balancer or mobility manager) the STA is immediately handed over to another AP.

4.3 STA Handover

In case a STA is not associated with proper AP, i.e. due to load balancing and mobility issues, a handover operation is performed by the BIGAP controller. As an illustrative example Fig. 4.1 shows the required steps to perform a handover of *STA* from *AP1* to *AP2*:

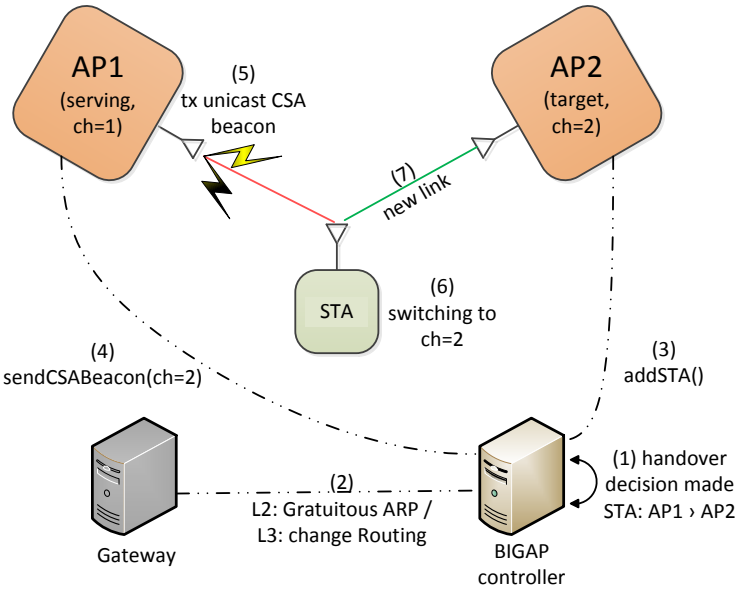

Figure 4.1: **The BIGAP handover.** Client STA is switched from AP1 to AP2.

i) A decision was made in the BIGAP controller to handover *STA* from *AP1* to *AP2*.

ii) The traffic flows towards *STA* need to be routed over *AP2*. There are multiple solutions to achieve this depending on whether bridging (sending a gratuitous proxy ARP message [16]) or routing (changing routing entry in gateway) is used.

iii) The BIGAP associates *STA* on the target AP, *AP2*, using the information about *STA* (e.g. client capabilities) provided by *AP1*. This make sure that after the handover operation the *STA* is properly registered[1] within *AP2* since otherwise *AP2* would respond with an disassociation frame and will not accept data frames.

iv+v) BIGAP controller instructs *AP1* to send out an unicast beacon containing a CSA-IE with the channel set to the target AP, here 2, destined to *STA*.

vi) On successfully receiving the unicast beacon containing the CSA the corresponding *STA* performs channel switching as specified in the IEEE 802.11 standard.

vii) Since both *AP1* and *AP2* have the same BSSID, aka MAC address, the *STA* does not notice that it is being served after the channel switch by another AP, *AP2*. *STA* continues with its communication.

[1] This step includes a successful 802.11 authentication and association plus a successful 802.1X authentication

Table 4.1: BIGAP AP API description.

ConfigManager	AP receives configuration commands from the BIGAP controller and executes them.
	configureAP(BSSID, SSID, rfChannel): sets the BSSID, SSID and the RF channel on the interface used in AP mode. *sendCSABeacon(STA, channel)*: sends an unicast beacon containing a CSA with channel of the target AP on the same channel where the serving AP is operating to the STA which should perform the handover. *addSTA(STA)*, associate STA on the AP where the function is called (target AP)
WirelessStats	AP reports wireless statistics to the BIGAP controller.
	getAPScanResults(): Scans the whole spectrum for neighboring APs beacons using the scanning interface and reports their SSID, BSSID (MAC address) and used RF channel. *getSTAScanResults()*: Scans the whole spectrum for neighboring STAs to-DS frames (data and management frames) using the scanning interface and reports their MAC address, average SNR and used RF channel. *getClientInfo()*: Reports information about associated STAs, i.e. MAC address, average SNR and inactivity time. *getTrafficInfo()*: Reports information about the aggregated airtime utilization of a WiFi channel to capture channel occupancy due to WiFi and non-WiFi activity at the AP's location.

4.4 BIGAP API

The API provided by each BIGAP AP is shown in Table 4.1. The API is used by the BIGAP controller to receive wireless statistics from each AP under control as well as to execute BIGAP commands (e.g. register new STAs in APs and perform handover operation). The actual function to perform a handover is defined as follows (Algorithm 1). First, the channel of the handover target AP need to be determined. Therefore, the BIGAP controller exploits the fact that within a single collision domain a particular channel is used at most once. Hence, by selecting a channel we implicitly select the corresponding AP. Second, the STA to be switched is registered in the target AP. Third, the controller triggers the transmission of the beacon packet containing the CSA-IE in the serving AP.

Algorithm 1 Function to perform a handover of a given STA from a serving AP to a target AP.

Require: STAaddr ▷ MAC address of STA to be switched.
Require: SrcAPId ▷ The ID of the handover source AP, i.e. the IP address of the wired backhaul interface.
Require: DstAPId ▷ The ID of the handover target AP.

```
1: procedure PERFORMHO(STAaddr, SrcAPId, DstAPId)
2:     c ← getChannelUsedbyAP(DstAPId)
3:     BIGAP::addSTA(DstAPId, STAaddr)
4:     BIGAP::sendCSABeacon(SrcAPId, STAaddr, c)
5: end procedure
```

Chapter 5

BIGAP – Implementation Details

Next we present details of the BIGAP prototype implementation.

5.1 BIGAP APs

The task of the BIGAP APs is twofold. On the one hand they enable the BIGAP controller to retrieve wireless statistics and on the other hand they allow him to control the behavior of the APs by executing configuration commands. The BIGAP prototypical implementation uses for the AP standard x86 machines with *Ubuntu 14.04.2 LTS* as operating system and *Linksys AE1000* WiFi USB sticks using Ralink rt2800 chipsets for the two wireless interfaces. To provide AP functionality the BIGAP APs run a modified version of *hostapd* [15] software in version 2.1. The API defined in Table 4.1 is implemented by the BIGAP APs as Remote Procedure Call (RPC) functionality by relying on the *ZeroRPC* [18] Python package which provides RPC on top of *ZeroMQ* [12]. Implementation details of each function are given below:

- *addNewSTA(mac, staCapa)* To realize the handover functionality, the BIGAP controller has to copy the current STA related state from the serving AP and install it on the target AP. We implemented this functionality by providing a hostapd_cli function [1] that calls the appropriate functions within hostapd which would also be triggered by a standard STA association (authentication, association and 802.1X authentication). The BIGAP controller calls this function on the target AP of a handover operation. Currently the implementation does not support any kind of security and only performs open system authentication.

- *sendCSABeacon(bssid, mac, channel)* After the client's state has been transferred to the target AP, the STA has to switch from its current channel to the channel used by the target AP. This is achieved by injecting an unicast beacon frame containing the CSA-IE with the channel of the target AP into the radio interface running in AP mode. We used the *Scapy library* [4] for frame construction and RadioTap [2] for frame injection.

- *getSTAScanResults()* This function is periodically called by the BIGAP controller on each AP under control. The function delivers all detected STAs together with link metrics like average SNR. Internally, every AP uses its second physical WiFi interface which is operating in monitor mode (cf. Fig. 3.1) to collect all 802.11 management and data frames that are destined to the distribution system. This process is performed periodically over all available channels. We used the *C++ libtins library* [9] for the frame processing. The results are stored within a ring

[1] hostapd_cli is a command-line interface which enables controlling of hostapd during runtime.

buffer. When the getSTAScanResults() function is executed, the mean SNR of all seen STAs is calculated over the last N collected frames that are not older than a predefined amount of time, e.g. 3 s.

- *getAPScanResults()* During the bootstrapping process, the BIGAP controller executes this function on the new AP to get an local view of neighboring APs. The function also uses the second WiFi interface operating in monitor mode for passive scanning. In contrast to the *getSTAScanResults()* function, only beacon frames are collected and parsed using the libtins library. For each neighboring AP the average SNR together with the used channel is reported back to the BIGAP controller.

- *getClientInfo()* Makes use of the *iw tool* [3] to provide information about associated STAs like the average SNR and the STA inactivity time. The latter is used by the BIGAP controller to find out by which AP a particular STA is currently served, i.e. the AP with the smallest STA inactivity time.

- *getTrafficInfo()* Using the *C++ libtins library* [9] the aggregated airtime utilization of a channel was calculated from received 802.11 frames by also taking into account of inter-frame spaces and the contention window.

- *configureAP(BSSID, SSID, rfChannel)* The bootstrapping process automatically configures the needed parameters for an BIGAP AP when it is newly integrated into the network. This function enables the BIGAP controller to setup the AP parameters (SSID, BSSID) and the automatic startup of hostapd. Moreover, it also supports the reconfiguration of already running BIGAP APs during runtime for channel assignment.

5.2 BIGAP Controller

The BIGAP APs are controlled by the central BIGAP controller which makes use of the RPC functions provided by the BIGAP APs. All AP nodes are automatically discovered by the BIGAP controller using the ZeroMQ Realtime Exchange Protocol (ZRE) [11]. The controller is also implemented in Python and uses the ZeroRPC library for executing RPC calls on the BIGAP nodes. When routing instead of bridging is used in the wired backhaul, the controller is also responsible for updating the routing table in the gateway by utilizing the Netlink API.

Chapter 6

BIGAP's Applications

Next we present three applications supported by BIGAP.

6.1 Load Balancing Clients

The objective is to optimize client associations across APs which is useful when AP radio interfaces become overloaded with traffic, e.g. full hotspot cell with lightly loaded neighboring cells. This requires information sharing about client load, airtime utilization, CRC error rates, and RF interference conditions. The BIGAP controller uses our specified interface to get this information from the BIGAP APs and to stores it in a local database of radio and network conditions. This allows the BIGAP controller to handover STAs to APs having the most favorable performance conditions. In particular we perform an airtime-based load balancing where the load is measured based on the channel utilization. More advanced algorithm may also take external interference on the channel into account.

6.2 Seamless Mobility

In a mobile scenario a handover is required because clients leave coverage of one cell (AP) and enters coverage of another AP. This requires information sharing about the quality of the currently used wireless link, i.e. SNR of link between client and serving AP, as well as the quality of links to candidate APs. The latter information is obtained using the second scanning interface in each BIGAP AP. Both information is collected by each AP and offered to the BIGAP controller which can trigger the handover operation. BIGAP uses the link SNR value as metric.

6.3 Interference Management

Hidden and exposed node problems [21] are known issues in WiFi networks which result either in interference (collisions) or inefficient use of the channel. The handover of a client to another AP operating on a different channel is a promising way to combat those problems. In particular the global view of the BIGAP framework allows the implementation of algorithms for hidden and exposed node detection and to apply handover of clients to mitigate this problem. Especially in the envisioned WiFi network with high AP density there is a multitude of candidate APs for handover.

Chapter 7

Evaluation

The objective of this section is to evaluate the proposed BIGAP architecture. At first we present the experimental methodology. Thereafter the results are presented and discussed.

7.1 Methodology

BIGAP is analyzed by means of experiments in a small 802.11n/ac testbed. The hardware used for the APs was already described in §5.1. For the clients we used two unmodified Android smartphones, i.e. Samsung Galaxy Note 2 and 3 running Android 4.4.2 and 5.0 respectively. For the experiments we used two unused channels from the 5 GHz band, i.e. channel 40 and 44.

We considered the following three experiments. First, we wanted to analyze the cost of the proposed handover scheme in terms of outage duration and throughput degradation during the handover operation. Second, we present results from a load balancing experiment. Third, we analyzed the support for seamless mobility.

7.2 Results

Experiment 1: (Analyzing the handover cost) The objective is to analyze the cost of the proposed BIGAP handover scheme in terms of outage duration and throughput degradation and to compare it with a hard handover scheme as proposed by Murty et al. [16]. Therefore, the following experimental setup was selected (Fig. 7.1). Two BIGAP APs and a single STA were placed close to each other ($\approx 1\,m$) to ensure very high link quality. The BIGAP controller was configured to perform a periodic handover of the STA between the two APs.

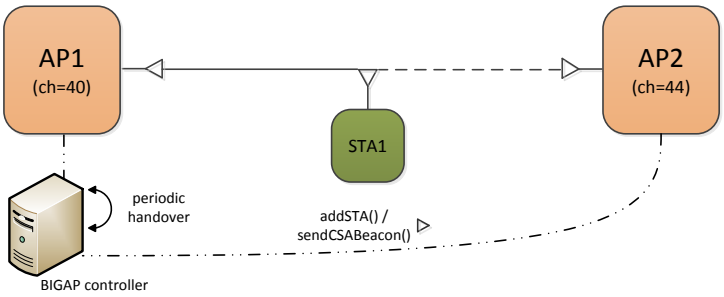

Figure 7.1: Experiment setup for seamless handover with BIGAP.

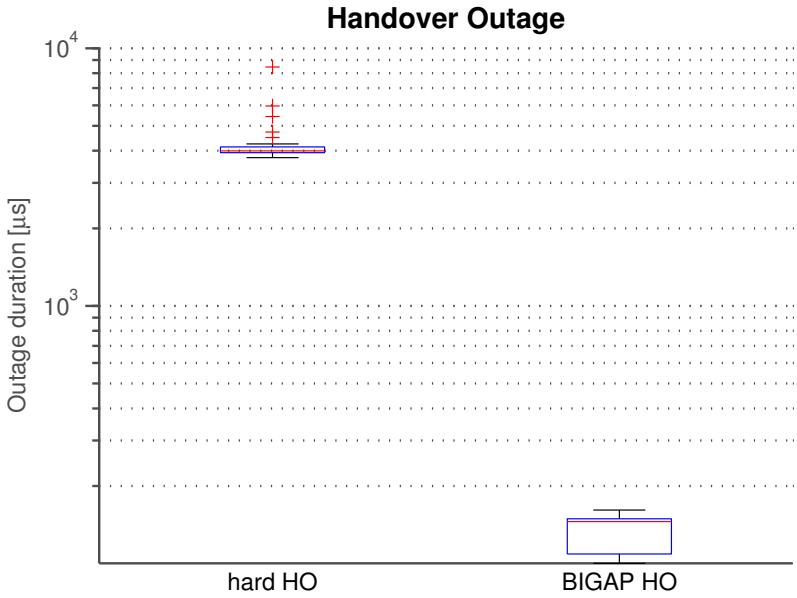

Figure 7.2: Network outage duration due to handover.

Result 1: At first we were interested in measuring the outage duration due to the handover. Therefore, we set up an ICMP ping flow from some server in the backhaul to the STA. The ping interval was set to 10 ms. We recorded all received ICMP reply packets on the server and measured the inter-arrival time between consecutive ICMP replies. The results are shown in Fig. 7.2. We can clearly see that the outage duration of the proposed handover scheme is on average 32 $\times$ smaller as compared with a hard-handover scheme, i.e. 0.13 s vs. 4.26 s.

Next, we evaluated the impact of the handover operation on a TCP/IP flow. We setup a TCP/IP flow using iperf [22] from the server towards the STA (downlink). Fig. 7.3 shows the throughput averaged over 50 runs. We can see that there is a slightly degradation in throughput during the HO operation of around 5 %. Note, TCP/IP is sensitive to packet reorder and loss which happens during a handover. The focus of this paper is the handover in the wireless access part. The performance can be improved when optimizing also the handover in the backhaul.

Experiment 2: (Load Balancing) The objective is to show the advantage from seamless load balancing. Fig. 7.4 shows the experiment set-up. Here we have two STAs which are initially associated to AP1. STA1 was again our smartphone whereas STA2 was a Linux laptop with an Intel 802.11ac chipset. Because both APs operate on different channels a load balancing is meaningful, i.e. STA1 is switched to AP2. Similar to the previous experiment we setup a TCP/IP flow one for each STA. The BIGAP controller was configured to perform load balancing.

Result 2: Fig. 7.5 shows the TCP throughput of STA1 before and after handover averaged over 50 runs. We see that the throughput is increased by 4 $\times$ after the handover of STA1 to AP2. Here STA1 was able to use the whole channel alone and also experienced a better SNR.

Experiment 3: (Seamless Mobility) The objective is to show that BIGAP supports seamless mobility. BIGAP is able to support this by using the second air interface in the AP for STA detection. We implemented a simple mobility scheme where the handover is performed based on SNR, i.e. the STA

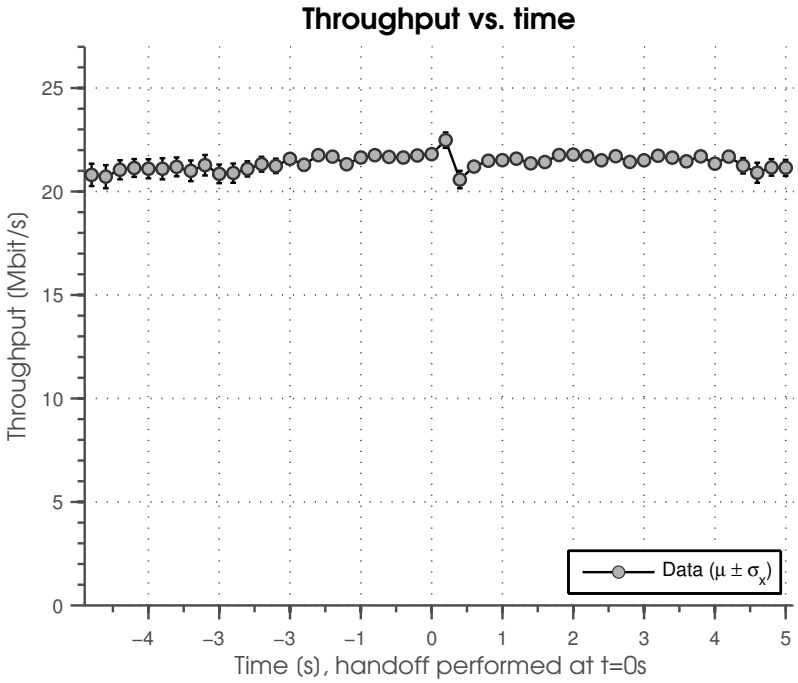

Figure 7.3: TCP/IP throughput over time. At t=0 s the handover operation was performed. The mean and standard error value is presented.

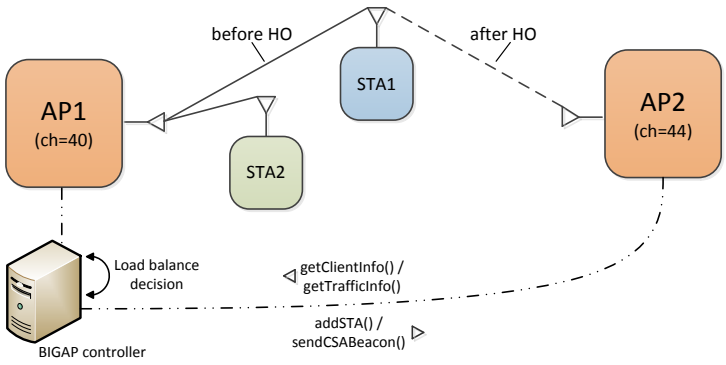

Figure 7.4: Experiment setup for load balancing with BIGAP.

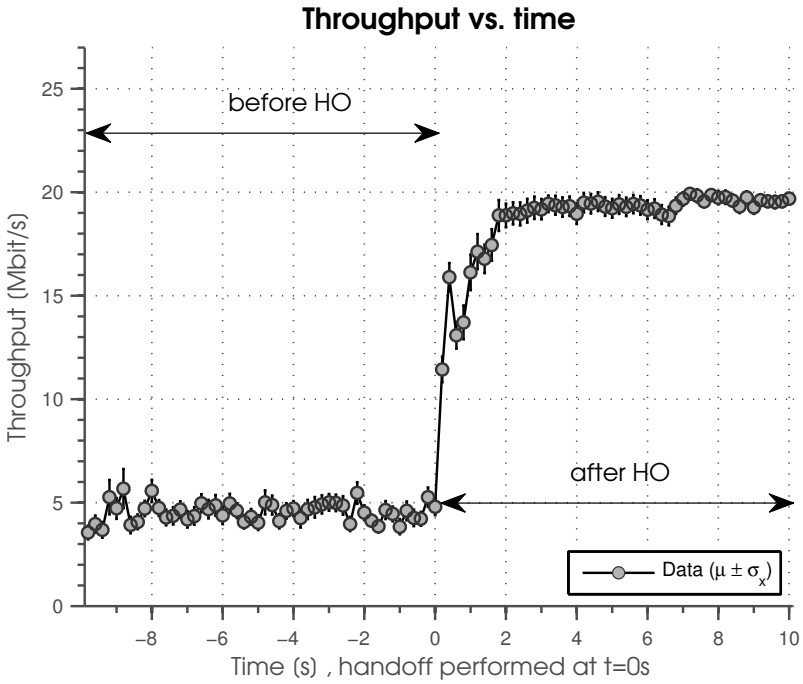

Figure 7.5: TCP/IP throughput of STA1 before and after Load balancing with BIGAP.

is switched to the AP to which it has the highest link SNR. To avoid the ping-pong effect we used a hysteresis value of $\tau = 8dB$. The experiment setup is shown in Fig. 7.6. Two APs were placed at a distance of 34 m. The STA was moving at a constant speed of 1 m/s between the two APs. Moreover, we setup a single TCP flow from the server in backhaul towards the STA.

Result 3: The results are shown in Fig. 7.7. In addition to the TCP throughput we also show the points in time where BIGAP performed a handover operation.

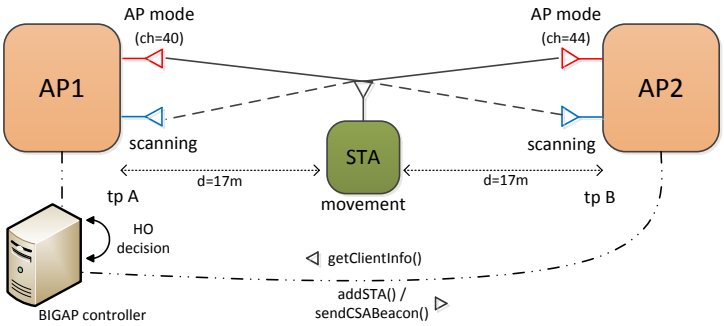

Figure 7.6: Experiment setup for seamless mobility with BIGAP.

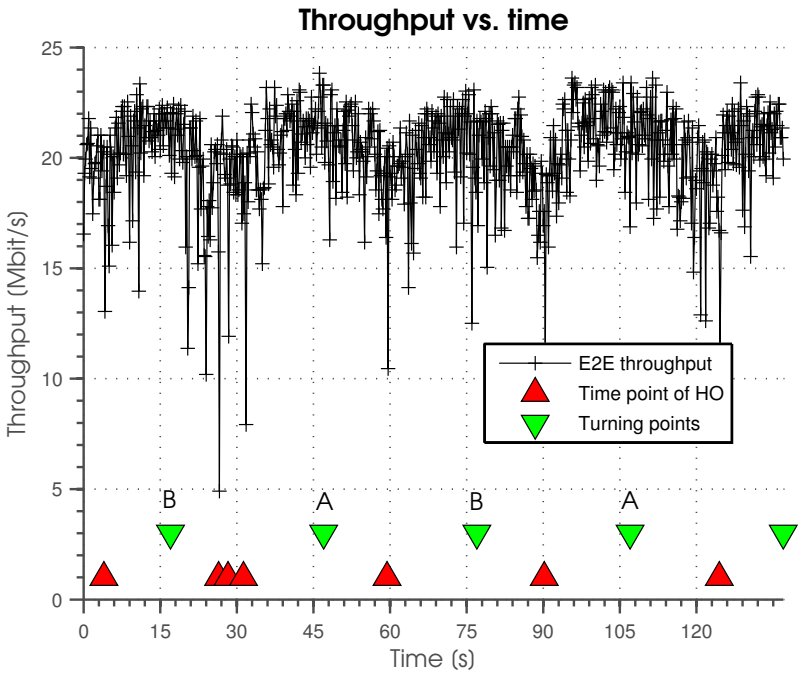

Figure 7.7: Seamless mobility with BIGAP.

Chapter 8

Discussion

In this section we discuss the limitations and future improvements of the proposed solution.

8.1 Enhancement

To ensure seamless handover the use of 802.11 Block Acknowledgments (BA) is problematic. This is because of the statefulness of the protocol. A handover would confuse both the target AP as well as the STA. Moreover, the old AP would pollute the channel by performing unnecessary retransmissions of all not acknowledged frames in the transmission window (block ARQ). There are two options to solve this problem. First, is the complete switching off of BAs. Second, is a gracious tear down of a BA agreement just before the handover operation takes place and the tearing up on the target AP (cf. first column in Table 8.1). Since, currently there is no API available to control BAs BIGAP chooses option one whereas option two is for future research.

Although the 802.11 Standard Acknowledgment (SA) scheme is inefficient, it provides an advantage. Since SA frames are sent with the same bitrate, at the same time and also contain the same content transmit diversity at the physical layer can be exploited which improves the reliability of the acknowledgment frame [14].

A handover operation in the wireless access network requires also coordinated operations in the wired backhaul. In particular the flows belonging to the STA which is performing the handover operation need to be rerouted. If the flows are redirected too early, then the new AP will perform unnecessary retransmissions at the MAC layer due to STA deafness, i.e. STA is ether on the channel of the old AP or in the process of channel switching. Our idea is to instruct the target AP of the handover operation immediately after changing the routing in the backhaul that the newly registered STA is in power saving mode. This will result in buffering of packets destined to STA in the target AP. We prototypically implemented this functionality by patching the *ath9k* driver. Future research is on finding a proper API to enable the control from userspace.

8.2 Limitations

BIGAP assumes the existence of a sufficient large number of available channels to make a collision-free channel assignment to APs. In case of insufficient channels, e.g. due to unavailable DFS channels in 5 GHz band (radar), there are the following two options available for the normal operation (cf. first row in Table 8.1): i) to avoid retransmissions of uplink traffic due to collision of non-identical BA make use of less efficient SA together with duplicate elimination at the gateway node or ii) assign different BSSIDs to co-located APs on the same channel and restrict the handover between those to the hard handover scheme [16].

BIGAP does not consider security. For future work we plan to provide centralized WPA2 security by tunneling the encrypted 802.11 traffic.

Table 8.1: Recommended acknowledgment scheme for BIGAP.

Operation	Number of RF channels available	
	sufficient	insufficient
normal	BA	SA with duplicate elimination or BA with different BSSIDs
handover	SA	SA

Chapter 9

Related Work

In order to improve the performance of enterprise WiFi Murty et al. [16] proposed DenseAP, a WiFi network with dense deployment of APs where the decisions about channel assignment, and association of STAs is taken by a global controller. While the presented approach is able to provide high performance the proposed handover scheme which uses STA disassociation together with global blacklisting to perform AP initiated STA handover causes a severe outage during handover operation. Recently DenseAP was extended by Trantor [17] by adding support for controlling physical bitrates, transmission power, and clear-channel assessment.

To support seamless mobility in WiFi networks Grunenberger et al. [10] introduced the concept of Virtual Access Point (VAP), which is a mobile entity within the infrastructure network. Every mobile station is therefore associated with its own VAP when it connects to the network, the latter moving along with its client. Therefore, each client gets its own VAP with a unique BSSID. While VAP is able to provide seamless mobility this approach is due to the high overhead of handling VAP for each STA unscalable. Moreover, VAP requires all APs to operate on the same channel making it unsuitable for use in dense Enterprise WiFi Networks. Even in a moderate sized Enterprise WIFI (802.11n/ac) network with up to 6 co-located APs each with up to 8 STAs the management overhead due to VAP already consumes 25 % of the channel airtime making this solution practically infeasible (Fig. 9.1). In order to mitigate the large broadcast traffic overhead for managing VAP Yiakoumis et al. [23] proposed to increase the beacon interval which, however, might be problematic because of STA power management which relies on shorter beacon interval.

To take advantage of APs operating on multiple channels Multichannel Virtual Access Points (mVAP) was developed [1]. Similar to our approach mVAP makes use of the CSA-IE to force the STA to switch to the channel of the target AP. Since each STA gets its own VAP a broadcast beacon containing the CSA can be used. Regarding scalability the same applies as with single channel VAP; it does not scale with the number of STAs. In a moderate sized Enterprise WiFi and sufficient available channels the overhead with 20 STAs per AP is already around 10 %.

In contrast in BIGAP the management overhead is always constant at around 0.5 % and does not depend on the number of STAs and/or AP density. Moreover, since the CSA element is send as unicast frame the BIGAP handover is very robust to packet loss due to interference and competition in highly loaded networks.

Finally, there are also proposals for seamless handover which require modifications to the 802.11 standard. With Flashback [7] a new physical layer was proposed which allows nodes to reliably send short control messages concurrently with data transmissions. This allows the time consuming association protocol to runs in parallel. Chan et. al. [6] proposed a 802.11 handover scheme where action frames containing CSA-IE were used to trigger handover. A new management frame was introduced which allows the changing of the stored BSSID within the STA.

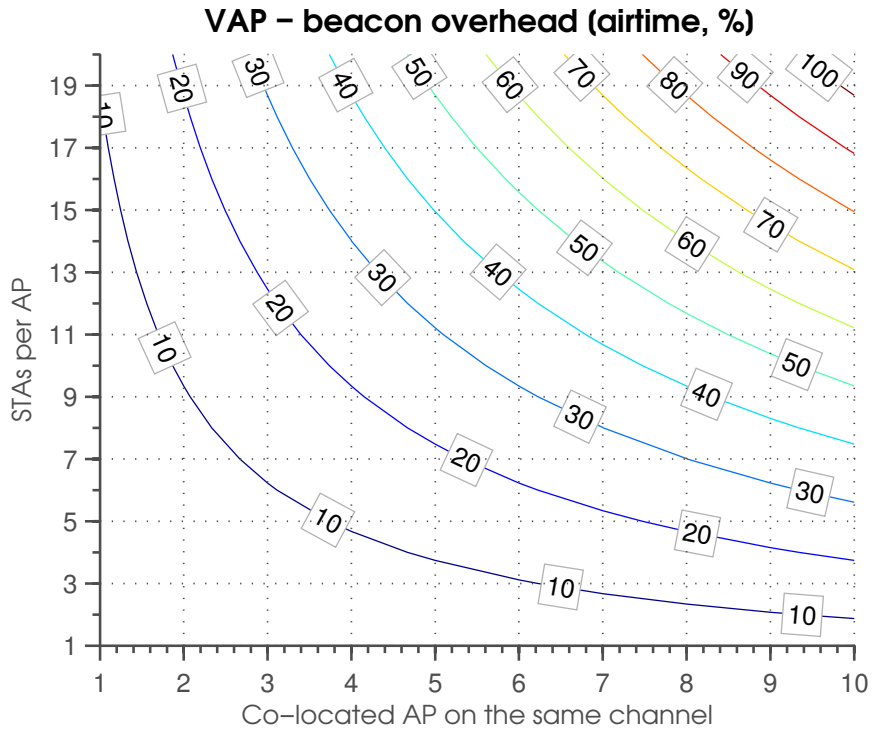

Figure 9.1: Wireless signaling overhead for managing of VAPs in 802.11a/g/n/ac.

Chapter 10

Conclusions

This paper introduces BIGAP which provides both high network performance as well as seamless handover in Enterprise WiFi networks. The former is achieved by fully utilizing the available radio spectrum whereas the latter is accomplished by providing a mechanism for below MAC-layer handover.

Bibliography

[1] Maria Eugenia Berezin, Franck Rousseau, and Andrzej Duda. Multichannel virtual access points for seamless handoffs in ieee 802.11 wireless networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.

[2] Johannes Berg. Radiotap. *http://www.radiotap.org/*, March 2011. Accessed: 2015-08-04.

[3] Johannes Berg. wireless configuration tool. *http://git.kernel.org/cgit/linux/kernel/git/jberg/-iw.git*, August 2015. Accessed: 2015-08-04.

[4] Philippe Biondi. Scapy. *http://www.secdev.org/projects/scapy/*, December 2014. Accessed: 2015-08-04.

[5] Sanjit Biswas, John Bicket, Edmund Wong, Raluca Musaloiu-E, Apurv Bhartia, and Dan Aguayo. Large-scale measurements of wireless network behavior. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 153–165. ACM, 2015.

[6] Yi-Cheng Chan and Dai-Jiong Lin. The design of an ap-based handoff scheme for ieee 802.11 wlans. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 4(1):72, 2014.

[7] Asaf Cidon, Kanthi Nagaraj, Sachin Katti, and Pramod Viswanath. Flashback: Decoupled lightweight wireless control. *ACM SIGCOMM Computer Communication Review*, 42(4):223–234, 2012.

[8] Cisco. The zettabyte era: Trends and analysis - whitepaper. *http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.pdf*, May 2015. Accessed: 2015-08-04.

[9] Matias Fontanini. libtins - packet crafting and sniffing library. *http://libtins.github.io/*, May 2015. Accessed: 2015-08-04.

[10] Yan Grunenberger and Franck Rousseau. Virtual access points for transparent mobility in wireless lans. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6. IEEE, 2010.

[11] iMatix Corporation. 36/zeromq realtime exchange protocol. *http://rfc.zeromq.org/spec:36*, January 2012. Accessed: 2015-08-04.

[12] iMatix Corporation. Zmq - code connected. *http://zeromq.org/*, January 2014. Accessed: 2015-08-04.

[13] Bastian Könings, Florian Schaub, Frank Kargl, and Stefan Dietzel. Channel switch and quiet attack: New dos attacks exploiting the 802.11 standard. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*, pages 14–21. IEEE, 2009.

[14] Mathias Kurth, Anatolij Zubow, and Jens-Peter Redlich. Cooperative opportunistic routing using transmit diversity in wireless mesh networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[15] Jouni Malinen. hostapd: Ieee 802.11 ap, ieee 802.1x/wpa/wpa2/eap/radius authenticator. *https://w1.fi/hostapd/*, January 2013. Accessed: 2015-08-04.

[16] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. Designing high performance enterprise wi-fi networks. In *NSDI*, volume 8, pages 73–88, 2008.

[17] Rohan Narayana Murty, Jitendra Padhye, Alec Wolman, and Matt Welsh. An architecture for extensible wireless lans. In *HotNets*, pages 79–84, 2008.

[18] Jérôme Petazzoni. Build reliable, traceable, distributed systems with zeromq (zerorpc). *http://-pycon-2012-notes.readthedocs.org/en/latest/dotcloud_zerorpc.html*, March 2012. Accessed: 2015-08-04.

[19] Stéphane Pomportes, Anthony Busson, Joanna Tomasik, and Véronique Veque. Resource allocation in ad hoc networks with two-hop interference resolution. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6. IEEE, 2011.

[20] Vikas Sarawat, Bernard McKibben, and Neeharika Allanki. Wireless access point load balancing, December 13 2014. US Patent App. 14/569,669.

[21] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. Centaur: realizing the full potential of centralized wlans through a hybrid data path. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 297–308. ACM, 2009.

[22] Ajay Tirumala, Feng Qin, Jon Dugan, Jim Ferguson, and Kevin Gibbs. Iperf. *http://web.-archive.org/web/20081012013349/http://dast.nlanr.net/Projects/Iperf/*, March 2003. Accessed: 2015-08-04.

[23] Yiannis Yiakoumis, Manu Bansal, Adam Covington, Johan van Reijendam, Sachin Katti, and Nick McKeown. Behop: a testbed for dense wifi networks. In *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 1–8. ACM, 2014.