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
Department of Telecommunication Systems, Technische Universität Berlin

Abstract—**Enterprise IEEE 802.11 networks need to provide high network performance to operate a large number of diverse clients like laptops, smartphones and tablets as well as capacity hungry and delay sensitive novel applications like mobile HD video & cloud storage efficiently. 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 achieving both of the above goals. The former is achieved by assigning different channels to co-located APs in order 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 802.11. In essence BIGAP forces clients to change AP whilst they 'believe' they are simply changing channel. BIGAP is fully compatible with 802.11 and requires no modifications to the wireless clients. Testbed results demonstrate a significant improvement in terms of network outage duration (which is $32\times$ smaller as compared to state-of-the-art solutions) and negligible throughput degradation during handover operation. In this way frequent and seamless handover operations can take place thus supporting both seamless mobility and efficient load balancing.

Index terms— Wireless, SDN, Handover, Mobility

I. Introduction

There is a clear trend towards deploying IEEE 802.11 wireless networks (WiFi) in enterprise environments. Moreover, the way enterprise WiFis are used has changed dramatically. WiFi enterprise customers would like to enjoy mobility indoor and outdoor. The appearance of WiFi enabled laptops, 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 [1]. In particular there are up to 25 non-overlapping channels available. Although STAs in a dense WiFi network can choose from many possible APs, this degree of freedom is not fully exploited in 802.11. 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. However, slower STAs are monopolizing airtime and hence significantly decreasing network capacity. 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 [2]. Therefore, an infrastructure-initiated handover scheme which allows better client load balancing is of fundamental importance in enterprise WiFi networks. In addition such mechanism might also support client mobility - a feature dramatically missing in today's networks, where switching to another AP requires significant service break. 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 however of integral importance in order to provide uninterruptible network connections and high QoS.

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 handover for mobility management and load balancing. While approaches like Virtual Access Point (VAP [3]) and mVAP [4] 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 [5] 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).

II. IEEE 802.11 Primer

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

A. 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 comprises a single AP and multiple associated STAs. If multiple APs are connected to a distribution system, the entire system with all single BSSes interconnected is called an Extended Service Set (ESS). All APs in an ESS are using the same Service Set Identifier (SSID) which serves as the network name visible to stations.

B. Handover in Standard 802.11

Handover operations 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 stickiness [2] 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 a 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 Address Resolution Protocol (ARP) cache or routing changes 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.

C. Channel Switch Announcement

The majority of channels in the 5 GHz band require a mechanism termed Dynamic Frequency Selection (DFS). 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 new channel to be used. 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, i.e. action frames.

III. BIGAP's Design Principles

Enterprise WiFi architectures need to be optimized to support a large number of STAs while providing high QoS, e.g. throughput and latency requirements. To fully utilize the available radio spectrum at each location a high density deployment of APs is required [5]. However, next generation enterprise WiFi networks also require strong support for mobility management, client load balancing and interference management. Hence, there is a need for a seamless client 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 [5]. 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 re-association. BIGAP decreases the network outage duration and removes all aforementioned delays by transferring the current state of the STA from the serving AP to the target AP. To enable this feature, BIGAP 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 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 channel switching in client device.

BIGAP does not require any modifications to the 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. This is feasible because there are enough channels available in the 5GHz band (25 with DFS).

BIGAP's general framework consists of two parts, cf. Fig 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

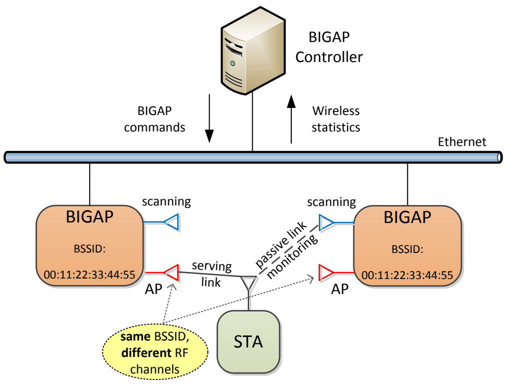

Fig. 1. **The BigAP architecture.** The BIGAP controller coordinates the APs using the BIGAP interface. All BIGAP APs of the same network have the same BSSID. Co-located APs are operated on different channels.

which uses wireless statistics like average link quality and traffic conditions.

BIGAP requires each AP to be 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 a beacon frame containing a CSA-IE with the RF channel of the target AP. Receiving this CSA-IE triggers the channel switching in the STAs to the desired RF channel. Since 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 [6]. BIGAP solves this problem by sending a 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.

IV. BIGAP - Detailed Specification

Next we give a description of the BIGAP specification.

A. AP Channel Assignment

When a 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 $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of BIGAP APs and there is an edge $e \in \mathcal{E}$ 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 [7]:

$$\forall v \in \mathcal{V} : \mathrm{ch}(v) \neq \mathrm{ch}(w), w \in \mathrm{nb}(\mathrm{nb}(v)) \tag{1}$$

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

B. 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 policy or algorithm running in the BIGAP controller the STA is immediately handed over to another AP using the scheme described in the next section.

C. STA Handover

In case a STA is not associated with a proper AP, i.e. due to load balancing, interference and mobility issues, a handover operation is performed by the BIGAP controller. As an illustrative example Fig. 2 shows the required steps to perform a handover of *STA* from *AP1* to *AP2*:

1) A decision was made in the BIGAP controller to handover *STA* from *AP1* to *AP2*.
2) The traffic flows towards *STA* need to be routed over *AP2*. There are two solutions to achieve this: i) bridging (sending a gratuitous proxy ARP message [5]) or ii) routing (changing routing entry in gateway).
3) The BIGAP controller associates and authenticates *STA* on the target AP, *AP2*, using the information about *STA* provided by *AP1*. This make sure that after the handover operation the *STA* is properly registered within *AP2* since otherwise *AP2* would respond with an disassociation frame and will not accept data frames.

4-5) 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*.

6) On successfully receiving the unicast beacon containing the CSA-IE the corresponding *STA* performs channel switching as specified in the IEEE 802.11 standard.
7) 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.

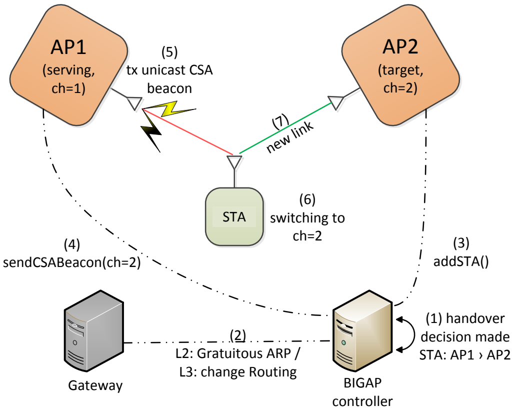

Fig. 2. **The BIGAP handover.** Client STA is switched from AP1 to AP2.

D. BIGAP API

The API provided by each BIGAP AP is shown in Table I. 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 Handover function used by BIGAP controller.

```
Require: STAaddr          ▷ MAC address of STA to be switched.
Require: SrcAPId ▷ The ID of the handover source AP (wired IP address).
Require: DstAPId          ▷ The ID of the handover target AP.
procedure PERFORMHO(STAaddr, SrcAPId, DstAPId)
    $c \leftarrow$ getChannelUsedbyAP(DstAPId)
    BIGAP::addSTA(DstAPId, STAaddr)
    BIGAP::sendCSABeacon(SrcAPId, STAaddr, $c$)
end procedure
```

V. BIGAP – Implementation Details

Next we present details of our BIGAP prototype implementation.

A. 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 LTS* 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* [8] software in version 2.1. The API defined in Table I is implemented by the BIGAP APs as Remote Procedure Call (RPC) functionality by relying on the *ZeroRPC* [9] and *ZeroMQ* [10]. 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 to 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 and association). 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 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* [11] for frame construction and injection.
- *getSTAScanResults()* This function is periodically called by the BIGAP controller on each AP under control. The function delivers all detected STAs together with wireless statistics like the mean link SNR. Internally, every AP uses its second WiFi interface which is operating in monitor mode (cf. Fig. 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 *libtins library* [12] for the frame processing. The results are stored within a ring buffer.
- *getAPScanResults()* During the bootstrapping process, the BIGAP controller executes this function on the new AP to get a local view of neighboring APs. The function uses the second wireless 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* [13] 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 *libtins library* [12] the aggregated airtime utilization of the channel is calculated from received 802.11 frames.
- *configureAP(BSSID, SSID, rfChannel)* The bootstrapping process automatically configures the required parameters for a BIGAP AP when it is newly integrated into the network. This function enables the controller to setup up the AP parameters (SSID, BSSID). Moreover, it also

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

TABLE I
BIGAP AP API DESCRIPTION.

ConfigManager	AP receives configuration commands from the BIGAP controller and executes them.
	configureAP(BSSID, SSID, rfChannel): configures the BSSID, SSID and the RF channel on the interface used in AP mode. *sendCSABeacon(STA, channel)*: sends a unicast beacon containing a CSA-IE with the channel of the target AP. *addSTA(STA)*: associates and authenticates an STA on an AP, ie. handover target AP.

WirelessStats	AP reports wireless statistics to the BIGAP controller.
	getAPScanResults(): Scans the spectrum for APs beacons using the scanning interface and reports their SSID, BSSID and used channel. *getSTAScanResults()*: Scans the spectrum for neighboring STAs To-DS frames (data and management frames) using the scanning interface and reports their MAC address, average SNR and used 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.

supports the reconfiguration of already running BIGAP APs during runtime for channel assignment.

B. 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) [14]. 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 Linux Netlink API.

VI. BIGAP'S APPLICATIONS

Next we present three applications supported by BIGAP.

A. 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. hotspot cells. This requires information sharing about client load, airtime utilization and RF interference conditions. The BIGAP controller uses our specified API to get this information from the BIGAP APs and to store 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.

B. Seamless Mobility

In a mobile scenario a handover is required because clients leave coverage of one 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.

C. Interference Management

Hidden and exposed node problems [15] are known issues in WiFi networks which result in 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 effect. Especially in the envisioned WiFi network with high AP density there is a multitude of candidate APs for handover.

VII. EVALUATION

As the main objective of this work is providing a novel handover technique - we do not present any particular mobility or load balancing policy. Thus in the performance evaluation we focus on the widely adopted performance metrics, namely network outage duration and throughput degradation during the handover operation. As both metrics depend on the channel switching duration of the utilized WiFi chipset/driver which is constant and independent of the total number of client STAs and APs, we do not see a need in providing large scale measurements with large number of client STAs and APs.

A. 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 §V-A. 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 our experiments we operate on two unused channels from the 5 GHz band, i.e. channel 40 and 44. We considered the following three experiments. First, we 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.

B. 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

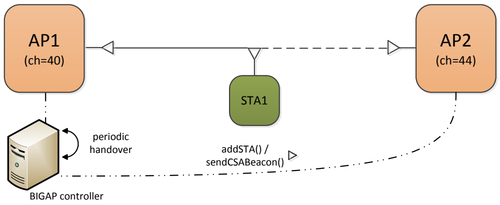

Fig. 3. Experiment setup for seamless handover with BIGAP.

Murty et al. [5]. Therefore, the following experimental setup was selected (Fig. 3). 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.

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 a server in the backhaul to the STA. The ping interval was set to 10 ms. From the recorded ICMP reply packets we measured the inter-arrival time between consecutive ICMP replies during the handover operation to obtain the outage duration with an accuracy of around 10 ms. The experiment was repeated to perform 50 handover operations for the soft BIGAP handover and the hard handover respectively. The results are shown in Fig. 4. We can clearly see that the outage duration of the proposed handover scheme is on average 32 × smaller as compared with a hard-handover scheme, i.e. 0.13 s vs. 4.26 s. In future, the BIGAP handover outage duration may further decrease, as Atheros and other WiFi chip manufacturers claim to be able to achieve switching times of 2 ms or less for their newer IEEE 802.11 chipsets [16].

Moreover, the two considered approaches can be compared with each other with respect to energy consumption in the client STA during the handover operation. According to the energy model proposed by Lin et al. [17] a full WiFi scan over the 11 radio channels in the 2.4 GHz band consumes around $343\,mJ$ for an already active WiFi chip. Hence, a scanning of the entire 2.4 and 5 GHz WiFi band, i.e. switching over 41 radio channels[2], would consume approx. $1277.56\,mJ$. As with the BIGAP soft-handover there is no need for a client STA to scan all channels, instead a single channel switch which is consuming around $31.16\,mJ$ ($\frac{1277.56\,mJ}{41}$) is sufficient. By way of illustration we calculated the number of times the handover operation can be performed before depleting the battery of a standard smart phone. With our test device Samsung Galaxy Note II with $3100\,mAh \approx 11.47\,Wh \approx 41,292\,J$, the entire battery would be exhausted after 32,321 hard and 1,325,160 soft-handover operations respectively.

Next, we evaluated the impact of the handover operation on a TCP/IP flow. We set up a single flow using iperf [18] from the server towards the STA (downlink). Fig. 5 shows the throughput averaged over 50 runs for both the proposed soft (upper) and the hard handover (lower). We can see that with the BIGAP soft-handover there is only a slightly degradation in throughput during the HO operation of just around 5 %. The situation is completely different with hard handover where the throughput drops to zero for almost 4 seconds.

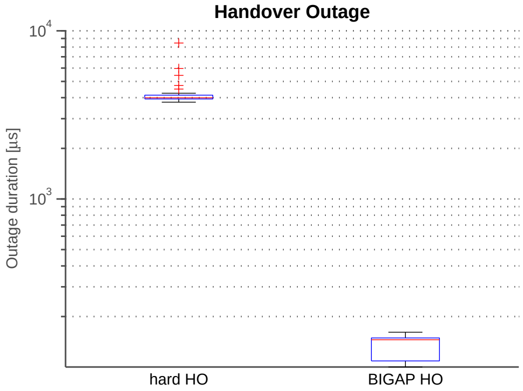

Fig. 4. Network outage duration due to handover.

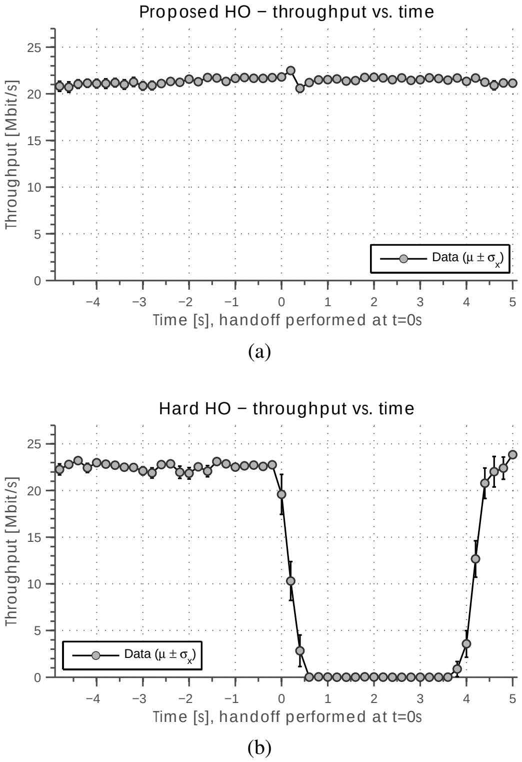

Fig. 5. TCP/IP throughput over time of proposed BIGAP soft-handover (a) and the standard hard handover (b). At t=0 s the handover operation was performed. The mean and standard error value is presented.

Experiment 2: (Load Balancing) Load balancing algorithms can be used to rapidly adjust the traffic load among different APs. Current load balancing schemes can achieve good load balancing performance, but due to the lack of a seamless soft-handover scheme, all of them also cause a constant connection outage, cf. Figure 4 and 5b. The objective of this experiment is to show that BIGAP supports seamless load-balancing and therefore provides a solution that preserves a good user

[2] Europe 802.11n regulatory domain 13 x 2.4 GHz and 28 x 5 GHz channel

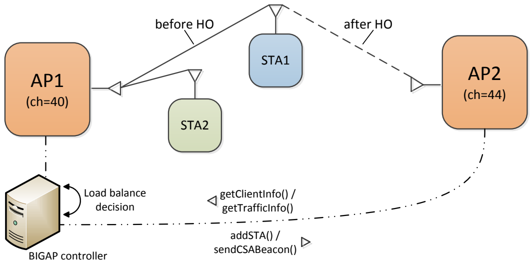

Fig. 6. Experiment setup for load balancing with BIGAP.

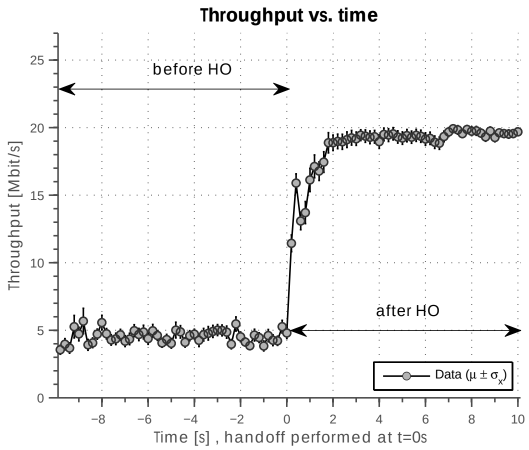

Fig. 7. TCP throughput of STA1 before and after Load balancing with BIGAP.

experience. Fig. 6 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 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 entire channel alone and also experienced a better SNR.

Experiment 3: (Seamless Mobility) The objective of this experiment is to show that BIGAP supports seamless mobility. BIGAP is able to detect the need for a handover by using the second air interface in the AP for the detection of STAs in proximity. We implemented a simple mobility scheme where the handover is performed based on SNR, i.e. the STA 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 $\tau = 8\,dB$. The experiment setup is shown in Fig. 8. Two APs were placed at a distance of 34 m. The STA was moved by an experimenter at a constant speed of $\approx 1\,m/s$ between the two APs. Moreover, we setup a single TCP flow from the server in backhaul towards the STA.

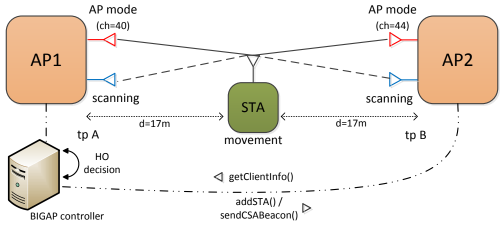

Fig. 8. Experiment setup for seamless mobility with BIGAP.

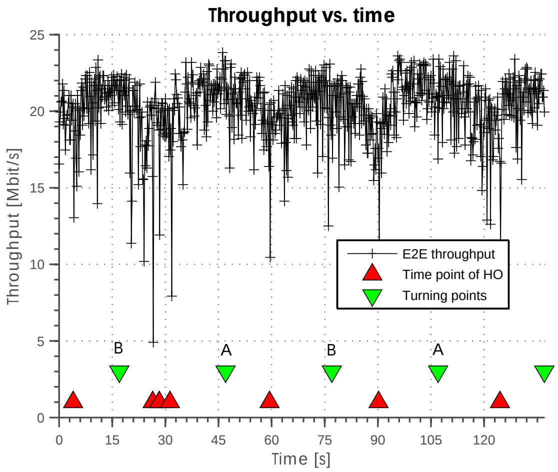

Fig. 9. Seamless mobility with BIGAP.

Result 3: The result from a single run is shown in Fig. 9. In addition to the TCP throughput we also show the points in time where BIGAP performed a handover operation. Note that around the 30s mark three consecutive handover operations occurred during a short time period.

VIII. Discussion and Conclusions

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

A. *Enhancement*

The use of 802.11 Block Acknowledgments (BA)[3] is problematic due to 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. There are two options to solve this problem. First, is to completely deactivate BA functionality. Second, is a gracious tear down of a running BA session on the serving AP just before the handover operation takes place and a new BA session initiation on the target AP after the handover is finished (cf. first column in Table II). Since, currently there is no API available to control BAs BIGAP chooses option one whereas option two is for future research.

A handover operation in the WiFi network requires also coordinated operations in the wired backhaul. In particular

[3] BAs enable the sender to send a stream of frames which can be acknowledged by the receiver using a single BA frame.

the packet flows belonging to the STA which is performing the handover operation need to be rerouted. Currently the BIGAP controller relies on IP routing (layer 3) while others, e.g. DenseAP [5] utilize layer 2 bridging. As future work, we aim to extend BIGAP with a SDN/OpenFlow extension as a way to steer the flows in the backhaul.

Nevertheless, independent of the utilized backhaul steering technique, the point in time when the flow steering is triggered is important. 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 either 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.

BIGAP does not consider security. We plan to provide centralized WPA2 security by tunneling the encrypted 802.11 traffic to a centralized security management unit. Using this approach, BIGAP has no impact on protocols like 802.1x and 802.11i. As the encryption and decryption functionality is handled centralized, all security related 802.11 functionality stays unmodified. From the perspective of the security management unit and from client STA perspective, it looks like the client STAs never performed a handover operation.

B. Limitations

BIGAP requires 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 II): i) to avoid retransmissions of uplink traffic due to collision of non-identical BA make use of less efficient Standard Acknowledgment (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 [5].

TABLE II
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

IX. Related Work

To improve the performance of enterprise WiFi Murty et al. [5] 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 network outage during handover operation. Recently DenseAP was extended by Trantor [19] by adding support for controlling physical bitrates, transmission power, and clear-channel assessment.

To support seamless mobility in WiFi networks Grunenberger et al. [3] introduced the concept of Virtual Access Point (VAP), which is a mobile entity within the infrastructure network. Every mobile STA 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 VAP management overhead already consumes 25 % of the channel airtime making this solution practically infeasible. In order to VAP management overhead Yiakoumis et al. [20] 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 VAP (mVAP) was developed [4]. 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-IE 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 [21] 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 run in parallel.

X. 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 exploiting 802.11 DFS functionality.

References

[1] S. Biswas, J. Bicket, E. Wong, R. Musaloiu-E, A. Bhartia, and D. Aguayo, “Large-scale measurements of wireless network behavior,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. ACM, 2015, pp. 153–165.

[2] V. Sarawat, B. McKibben, and N. Allanki, “Wireless access point load balancing,” Dec. 13 2014, uS Patent App. 14/569,669.

[3] Y. Grunenberger and F. Rousseau, “Virtual access points for transparent mobility in wireless lans,” in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[4] M. E. Berezin, F. Rousseau, and A. Duda, “Multichannel virtual access points for seamless handoffs in ieee 802.11 wireless networks,” in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, 2011, pp. 1–5.

[5] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill, “Designing high performance enterprise wi-fi networks.” in *NSDI*, vol. 8, 2008, pp. 73–88.

[6] B. Könings, F. Schaub, F. Kargl, and S. 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*. IEEE, 2009, pp. 14–21.

[7] S. Pomportes, A. Busson, J. Tomasik, and V. Veque, “Resource allocation in ad hoc networks with two-hop interference resolution,” in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[8] J. Malinen, “hostapd: Ieee 802.11 ap, ieee 802.1x/wpa/wpa2/eap/radius authenticator,” *https://w1.fi/hostapd/*, January 2013, accessed: 2015-08-04.

[9] J. 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.

[10] iMatix Corporation, “Zmq - code connected,” *http://zeromq.org/*, January 2014, accessed: 2015-08-04.

[11] P. Biondi, “Scapy,” *http://www.secdev.org/projects/scapy/*, December 2014, accessed: 2015-08-04.

[12] M. Fontanini, “libtins - packet crafting and sniffing library,” *http://-libtins.github.io/*, May 2015, accessed: 2015-08-04.

[13] J. Berg, “wireless configuration tool,” *http://git.kernel.org/cgit/linux/-kernel/git/jberg/iw.git*, August 2015, accessed: 2015-08-04.

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

[15] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. 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*. ACM, 2009, pp. 297–308.

[16] J. Herzen, R. Merz, and P. Thiran, “Distributed spectrum assignment for home wlans,” in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 1573–1581.

[17] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, “Energy-accuracy trade-off for continuous mobile device location,” in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’10. New York, NY, USA: ACM, 2010, pp. 285–298. [Online]. Available: http://doi.acm.org/10.1145/1814433.1814462

[18] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, “Iperf,” *http://web.archive.org/web/20081012013349/http://dast.nlanr.net/-Projects/Iperf/*, March 2003, accessed: 2015-08-04.

[19] R. N. Murty, J. Padhye, A. Wolman, and M. Welsh, “An architecture for extensible wireless lans.” in *HotNets*, 2008, pp. 79–84.

[20] Y. Yiakoumis, M. Bansal, A. Covington, J. van Reijendam, S. Katti, and N. McKeown, “Behop: a testbed for dense wifi networks,” in *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2014, pp. 1–8.

[21] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath, “Flashback: Decoupled lightweight wireless control,” *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 223–234, 2012.