CloudMAC - An OpenFlow based Architecture for 802.11 MAC Layer Processing in the Cloud

Peter Dely, Jonathan Vestin, Andreas Kassler
Computer Science Department
Karlstad University
Karlstad, Sweden
Email: peter.dely@kau.se, impclaw@gmail.com,
andreas.kassler@ieee.org

Nico Bayer, Hans Einsiedler, Christoph Peylo
Telekom Innovation Laboratories
Berlin, Germany
Email: {first.last}@telekom.de

***Abstract*—IEEE 802.11 WLANs are a very important technology to provide high speed wireless Internet access. Especially at airports, university campuses or in city centers, WLAN coverage is becoming ubiquitous leading to a deployment of hundreds or thousands of Access Points (AP). Managing and configuring such large WLAN deployments is a challenge. Current WLAN management protocols such as CAPWAP are hard to extend with new functionality. In this paper, we present CloudMAC, a novel architecture for enterprise or carrier grade WLAN systems. By partially offloading the MAC layer processing to virtual machines provided by cloud services and by integrating our architecture with OpenFlow, a software defined networking approach, we achieve a new level of flexibility and reconfigurability. In CloudMAC APs just forward MAC frames between virtual APs and IEEE 802.11 stations. The processing of MAC layer frames as well as the creation of management frames is handled at the virtual APs while the binding between the virtual APs and the physical APs is managed using OpenFlow. The testbed evaluation shows that CloudMAC achieves similar performance as normal WLANs, but allows novel services to be implemented easily in high level programming languages. The paper presents a case study which shows that dynamically switching off APs to save energy can be performed seamlessly with CloudMAC, while a traditional WLAN architecture causes large interruptions for users.**

***Index Terms*—Software Defined Networking, MAC Layer, Cloud**

I. Introduction

Wireless Local Area Networks (WLANs) operated by large companies or universities can consist of hundreds or even thousands of Access Points (APs). In such networks, administrators typically do not manage individual APs, but use enterprise WLAN management systems. Those systems hide the complexity of managing heterogeneous networks with different hardware and software platforms. Enterprise WLAN management systems often use protocols such as CAPWAP [1] and LWAPP [2] which allow vendor independent configuration of APs.

Besides the size of some WLANs, the configuration complexity of APs themselves is a challenge. Network virtualization, roaming support, quality of service and energy saving lead to increasingly complex APs. For example, the number of software packages included in the current release of the popular open source AP firmware OpenWRT increased by more than 800% over the past five years [3]. The operating system image size of the widespread Cisco 500 series APs increased by more than 150% from 2007 to 2011 [4].

Moving complexity away from networking devices and exposing data flow management functions via standardized interfaces and high level programming primitives is one of the core ideas of Software Defined Networking (SDN). OpenFlow [5], one example of SDN, has gained a lot of attention. For example, Google has replaced all its inter-data center backbone routers with simple OpenFlow-enabled devices, that are controlled from applications running on standard servers [6]. Thereby, the network flexibility can be increased while at the same time reducing the operational expenditure and the cost of network devices.

To get the similar benefits in WLANs, we introduce CloudMAC, a new management architecture in which APs just forward MAC frames. All other functionality, such as the processing of MAC data or management frames is implemented in standard servers that are operated in data centers and can be provided via cloud computing infrastructure. Openflow is used to manage the flow and transmission properties of MAC frames. Due to its open nature, OpenFlow enables the rapid development of novel services as we will demonstrate later in this paper. OpenFlow allows us to leverage the fast packet processing in hardware switches to implement centralized control plane functions for power and rate control (similar to CENTAUR [7]). Such fast, hardware based packet processing is getting more important as the PHY rates of upcoming IEEE 802.11 standards will reach multiple gigabits per second and pure software solutions are not capable of processing the traffic rates seen in centralized control planes. CloudMAC does not require any modifications to clients. A smooth transition from a traditional WLAN to a CloudMAC WLAN is possible, since the CloudMAC design supports all standard WLAN management tools. A preliminary performance evaluation shows that additional the processing overhead due to distributed MAC operation is small compared to the benefits of more flexibility.

The remainder of this paper is structured as follows. In Chapter II we introduce the architecture of the CloudMAC system, discuss implementation issues and highlight novel services and management approaches which become possible

978-1-4673-4941-3/12/$31.00 ©2012 IEEE

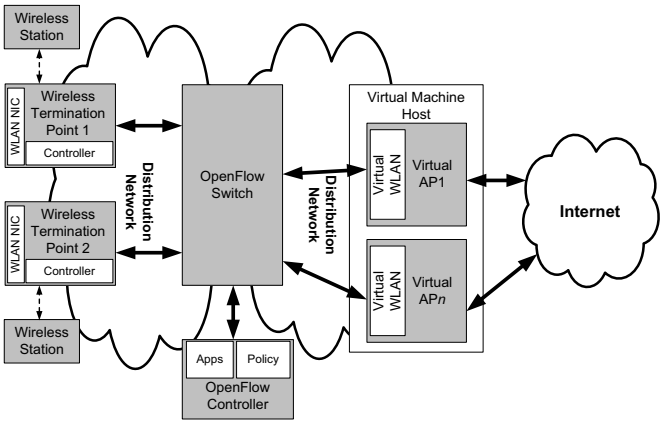


Figure 1: Architecture of a CloudMAC based WLAN

with CloudMAC. Chapter III presents preliminary evaluation results from real testbed measurements. Chapter IV contrasts CloudMAC with related work. Finally, in Chapter V we summarize the paper and provide an outlook to future work.

II. CloudMAC- An OpenFlow based Architecture for WLAN MAC Processing in the Cloud

A. CloudMAC Architecture

CloudMAC is a distributed architecture in which 802.11 WLAN MAC processing is partially performed in data centers on virtual machines connected by an OpenFlow controlled network. This is in contrast to traditional approaches, in which the complete MAC layer is located at the local AP. Our approach simplifies the management of WLAN deployments and allows a rapid deployment of new functionality using software modifications. A CloudMAC based WLAN deployment consists of Virtual APs (VAPs), Wireless Termination Points (WTPs), an OpenFlow switch, an OpenFlow controller and tunnels to connect the entities (Fig. 1).

VAPs are operating system (OS) instances running on a virtualization host, such as Xen or VSphere Center. Each VAP has one or several virtual WLAN cards. A virtual WLAN card is a driver that appears to the OS and user space applications like a normal physical WLAN card. Standard WLAN management tools can be used to set parameters of virtual WLAN cards. VAPs run access point management software, for example to generate beacon frames, or respond to Association/Authentication MAC frames. As the virtual WLAN card appears like a real card, standard software such as `hostapd` can be used. This allows having the increased flexibility through CloudMAC , while at the same time using well known standard software. One VAP can have many virtual WLAN cards, which are connected to physical cards on different WTPs. In the extreme case, one enterprise WLAN is only one VAP.

WTPs are slim APs with WLAN cards that allow to send and receive raw MAC frames. On the downlink, the virtual WLAN card in the VAP generates MAC frames, adds control headers and optionally encrypts the frames. WTPs transmit those frames to stations with the modulation/coding scheme and transmission power specified in the control header. The WTPs perform channel access using the Distributed Coordination Function [8]. Furthermore, WTPs generate frames with hard-real time constraints (ACKs and frame retransmissions). All other frames, such as beacons, are generated by the VAP. On the uplink, the WTP receives frames, acknowledges them and forwards them to the VAP for further processing.

VAPs and WTPs are connected with each other via layer 2 tunnels and an OpenFlow switch. The switch contains a switch table, which specifies what frames to forward to which WTP or VAP. The controller runs applications that configure the switch table using the OpenFlow protocol. The forwarding table represents the binding between VAPs (more specifically, the virtual WLAN cards) and WTPs. By reconfiguring the switch table, one AP can easily be moved from one WTP to another, together with all flows passing through it. As each physical card on a WTP can be bound to multiple virtual WLAN cards (traffic can be distinguished by the BSSID and MAC addresses), CloudMAC inherently supports network virtualization. Some OpenFlow implementations, such as OpenVSwitch [9], allow to re-write frame headers at arbitrary positions. This ability can be used to re-write the control header information, for example to implement centralized power or rate control.

Besides forwarding and modifying MAC data, control and management frames, CloudMAC allows fine grained control over configuration commands (CFG). Those are used to configure the WLAN card and typically issued by a user space application in a virtual AP. For example, if a user space application requests the virtual WLAN card to change its channel, the request is intercepted in the virtual WLAN card driver. The driver then generates a special CFG packet and sends it to the OpenFlow switch. The OpenFlow switch sends this CFG to the OpenFlow controller. If the CFG command is permitted according to a user-configurable policy, the OpenFlow controller forwards the CFG command to a control application residing on the WTP. The control application executes the command and returns the execution status to the OpenFlow controller and the VAP.

B. CloudMAC Implementation Aspects

We have implemented a prototype of CloudMAC on the KAUMesh testbed [10]. The WTPs are Cambria GW2358-4 embedded devices with WLM54AG cards with an Atheros 5212 chipset. The WTPs run a stripped down version of OpenWRT Backfire and use ATH5K as WLAN driver. The WLAN card uses monitor mode, which allows to transmit and receive raw MAC frames. We extended the driver to transmit frames at the PHY rate specified in the radio-tap header generated by the VAP. The WTPs utilize the multi-BSSID feature provided by Atheros chipsets and many other WLAN cards: the WLAN chipset includes a hardware register that controls which MAC addresses are used by the card and hence which MAC frames are acknowledged. The register can be configured from the OpenFlow controller via a control daemon that is part of the WTP.

The VAPs are Debian 6.0 VMs on a VSphere Center installation. The VAPs use `hostapd` [11] (version 0.6) to generate beacon messages and to provide authentication and authorization services. The virtual WLAN card driver is based on the `mac80211_hwsim` driver [11], which is a software simulator of an IEEE 802.11 device used for testing MAC functionality and user space tools such as `hostapd`. We modified `mac80211_hwsim` in order to allow reading and injecting RAW IEEE 802.11 frames. The control header added by the virtual WLAN card adheres to the radio-tap format.

Packets between the VAP, the switch and the WTP are tunneled using the Capsulator tool from OpenFlow project [12]. As we have no hardware OpenFlow switch to our disposal, we use OpenVSwitch 1.3.0 instead. OpenVSwitch [9] is a OpenFlow-compatible software switch in the Linux kernel. The switch runs in a VM on the same VSphere installation and is controlled by custom made switch controller implemented in Python. To bind together a WTP and a VAP the switch controller configures a rule on the switch, so that traffic from is simply forwarded between the tunnels.

C. Possible Applications

CloudMAC enables a range of new applications, for example:

Dynamic Spectrum Use: In CloudMAC, one virtual WLAN card can be connected to several physical WLAN cards. This enables a scenario, where one WTP contains several physical cards, operating with the same MAC address, but on different channels. The physical cards periodically monitor channel utilization and provide this information to the network controller. If a station is currently using a channel with high external interference, the OpenFlow controller sends an IEEE 802.11h Channel Switch Announcement message to the station to instruct it to switch to a less used channel. The station does not experience any interruption, as it can continue to communicate on the new channel with another physical card of the same WTP. Since the station is associated with the VAP (and not the WTP), no re-association is required. This procedure does not require any modification on the client, as long as it supports IEEE 802.11h (mandated by IEEE 802.11a/n).

On-demand AP: In today's virtualized WLANs one AP might broadcast the SSID for dozens of networks. As each SSID requires one beacon frame, an AP might broadcast hundreds of beacon frames per second thereby reducing available capacity for data transfers. CloudMAC enables a scenario, where the OpenFlow switch per default does not forward beacon frames from a VAP to the connected WTP. When a new user arrives and sends a probe request (that sometimes includes the SSID of the desired network), an application on the OpenFlow controller inspects the probe request and dynamically enables the beacons. The new user now receives the beacon and can connect to the network. If the probe request does not include the SSID, historical usage data and the users MAC address can be used to identify the desired network. Thereby the number of beacon messages on the wireless medium can be reduced.

Downlink Scheduling: All traffic between the virtual APs and the WTPs passes through the OpenFlow switch. The switch hence can be used for downlink scheduling either by simple rate shaping as provided by OpenFlow or by time division. For time division scheduling, the OpenFlow controller instructs the switch to only forward the packet of one WTP, while putting the packets of interfering WTPs in a queue on the switch. After one time slot, the switch rules are changed, so that packets of another WTP are released from the queue and forwarded.

III. Performance Evaluation

A. Micro Benchmarks

The split MAC processing done by CloudMAC could lead to performance degradations. We hence compare the performance of CloudMAC with a normal WLAN AP (using the same hardware) that is connected to an IP router via FastEthernet.

Using ping and iperf we measured the round trip time (RTT) and the TCP throughput between a station and the VAP/IP router. Figures 2 and 3 show the ECDF of the round trip time and the throughput for different TCP segment sizes. When CloudMAC is used, the (min/median/max) RTT increases from (1.30/1.79/12.17) ms for a standard WLAN system to (1.60/2.28/14.62) ms. This is due to additional processing at the OpenFlow switch and due to delay added by the tunnels. However, our experiments showed that time critical MAC frames like association response messages are delivered fast enough to allow standard clients (we tested with Windows XP, Linux and MacOS X) to associate to the CloudMAC VAP. We verified that the time for the whole association procedure was similar on both systems (around 1.44 sec). While CloudMAC shows small additional latencies due to the tunneling/OpenFlow overhead, the processing time for such control frames is significantly reduced due to the more powerful processing at the VAP. In our testbed, the VAP and the WTPs are on the same LAN and hence the RTTs are small. However, one might envision a setup, in which VAPs are located in some remote data centers. In such a setup the RTTs might be too large though, as some MAC frames like association response and probe response messages need to be delivered to stations within a few milliseconds.

Similarly, the TCP throughput is decreased slightly due to the additional components added by CloudMAC. For large TCP segments the performance decreases by approx. 8.5%. This performance decrease is due to the tunnels, which in our current implementation run in user space and therefore require context switching. In future work we plan to investigate the possibility of using kernel-space tunnels, which should result in improved performance.

B. Case Study - Seamless AP Switch-off System

As a demo application we have implemented a seamless AP switch-off system. Such a system can be one component for energy efficient WLAN operation: The idea is to save energy

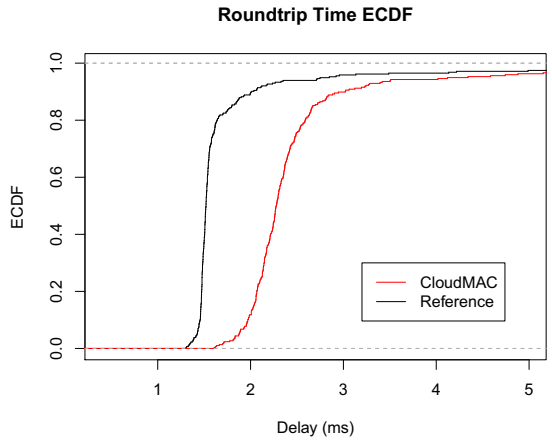


Figure 2: Ping RTT

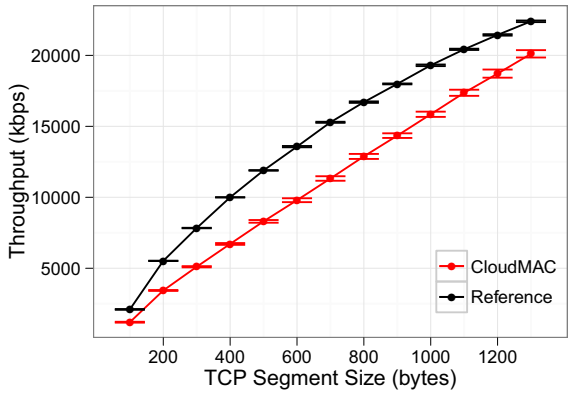


Figure 3: TCP throughput for different segment sizes

by switching off some APs/WTPs that cover overlapping area, if the network load is low [13]. In traditional WLANs such a system would require stations to re-associate to a new AP, which leads to a considerable service interruption. With CloudMAC the association state is kept in the VAP which allows to switch APs faster. Thereby WTPs can be switched off and on more frequently without disturbing ongoing connections and thereby even situation with temporarily small network load (e.g. lunch breaks) can be exploited. Our case study does not aim present a complete system for energy saving (which requires a scheduler that decides when to switch off which AP), but to show that CloudMAC is an enabler for seamlessly powering off APs.

Figure 4 depicts the test scenario, which in the CloudMAC case consists of two WTPs, one VAP and one standard IEEE 802.11 STA. The reference network consists of two regular APs (AP1 and AP2) that are connected to an external server via an Ethernet switch. The STA runs Linux and uses `wpa_supplicant` (version 0.8) [11] to find networks and associate to them.

In the first test, the STA connects to the network via WTP1 and the VAP/external server sends 1000 UDP datagrams per second to the STA. After a few seconds, the forwarding

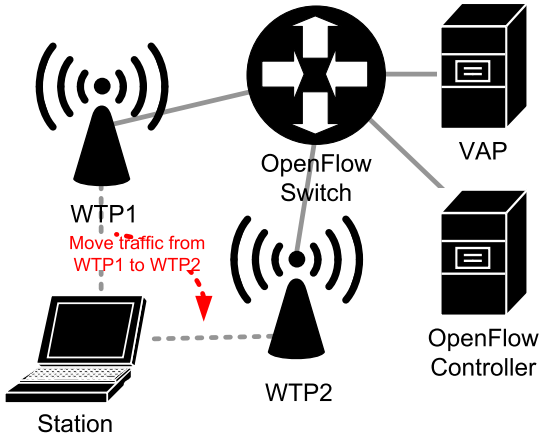


Figure 4: Test Scenario: first traffic is sent via WTP1 and then moved to WTP2

rules in the OpenFlow switch are changed and the register of MAC addresses is updated so that all traffic from WTP1 is moved to WTP2. Then WTP1 is switched off. In the reference network, the STA connects to AP1. After a few seconds `hostapd` sends a disassociation message to the STA and AP1 is switched off. `wpa_supplicant` searches for new APs and eventually connects to AP2, which then allows receiving UDP datagrams again. Figure 5 shows the Empirical Cumulative Distribution Function (ECDF) of the number of lost packets during the AP switch (20 trials). With CloudMAC on average 3.5 packets are lost (min: 2, max: 104). Those packets are in the transmit queue of the WTP during the AP switch and are lost during the switch. With the reference network, the AP/STA association is broken and the STA needs to find a new AP and connect to it. This results in much larger number of lost packets. On average 10780 packets are lost. Figure 5 also shows that the distribution of lost packets is bi-modal: In about 50% of the tests approximately 3200 packets were lost, while in the other 50% of the tests approximately 17400 packets were lost. This is because `wpa_supplicant` sometimes uses a cached list of available APs to select the next AP to associate to. Sometimes however, `wpa_supplicant` initiates a new scan for AP, which takes much longer time and results in the high number of lost packets.

In addition to finding and associating to a new AP, the dynamics of the transport layer can even cause a larger disruption to the user. We repeated the two tests with TCP downloads and show the corresponding time/sequence graphs in Figure 3. With CloudMAC, only a very small number of TCP segments gets lost during the switch and hence the time/sequence graph is smooth and steady. In the reference network though, there is no increase in sequence numbers (i.e. no throughput) between second 3 and 30. During that time, the link layer connection needs to be re-established and the TCP send rate needs to recover.

This shows that through CloudMAC APs can be switched seamless, while with a normal AP architecture this is not possible. We remark that with CloudMAC and standard IEEE

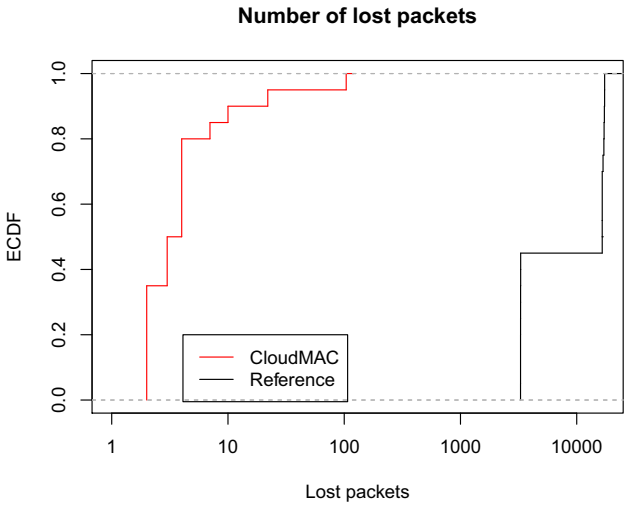
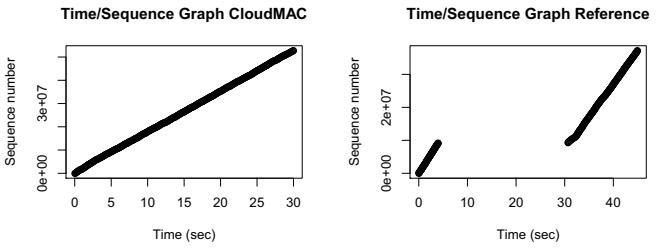


Figure 5: ECDF of number of lost packets during an AP switch



(a) CloudMAC (b) Standard IEEE 802.11 WLAN

Figure 6: TCP sequence-time diagram

802.11 STAs such a switch is even possible if the WTPs are operated on different channels. The IEEE 802.11h standard, which is mandatory for IEEE 802.11a/n devices, includes a Channel Switch Announcement (CSA) frame, in which the AP informs the STA to switch its channel. If WTP1 and WTP2 are using different channels, the OpenFlow controller just needs to generate such a CSA to request all STAs to switch channels before moving traffic from WTP1 to WTP2.

Moving an AP from one WTP to another WTP requires all STA associated to the AP to be within coverage of the new WTP. This might limit the situations in which a WTP can be switched off, since not all STA connected to one AP maybe in the coverage area of the new WTP. In the extreme case however, CloudMAC allows to use one AP per STA and hence APs can be moved in a very flexible way. Such a system could also be used to support station mobility.

IV. Related work

Related work falls into three main areas:

A. AP Virtualization

[14] and [15] proposed to run an OS hypervisor on APs and thereby provide virtual APs. Such full AP virtualization requires powerful APs (often also x86-based APs), which are not typical for normal WLAN deployments. Furthermore, [14] and [15] do not provide fine grained control of the transmission settings from within the virtual machine.

Besides full AP virtualization, virtual WLAN cards are now standard in Linux. Several virtual WLAN cards can be operated on top of one physical card. In contrast to our approach, a separation of administrative roles is not possible with virtual WLAN cards, as configuring a virtual WLAN card requires superuser privileges. In CloudMAC administrative roles can be separated by deploying e.g. one VAP for each virtual network.

B. OpenFlow Wireless

OpenFlow has been applied in the context of wireless networks in [16], [17], [18]. Those works focus on IP-layer management, on authentication and QoS-provisioning through OpenFlow. [16], [17] and [18] do not allow control over the MAC layer. CloudMAC is fully integrated into the OpenFlow architecture and can utilize OpenFlow infrastructure, such as switches, controllers and hypervisors. With Odin [19] the physical APs run software agents that provide authentication services and generate beacon frames. To enable a handover, Odin migrates the state of such an agent between APs using a self-defined protocol. OpenFlow is used to update forwarding tables in the network switches. CloudMAC significantly differs from Odin in the way association states are handled. While Odin keeps the state in the agents on the APs and a central controller, CloudMAC does not need to manage association states on the physical APs. This simplifies the design. Furthermore, the control headers in CloudMAC allow OpenFlow switches to control the characteristics of the wireless transmission, which is not possible with Odin.

C. Split MAC

As highlighted in the introduction of this paper, managing enterprise WLANs can be a difficult undertaking. In particular plain IEEE 802.11 APs have no capabilities for centralized network management. CAPWAP [1] and its predecessor LWAPP [2] are well established standards to address this problem. CAPWAP uses a centralized controller to discover APs, configure them and to provide authentication and authorization services to stations. CAPWAP implements a split MAC, in which some MAC frames are generated by the AP controller and others by the WTP. CAPWAP can be seen as a precursor to CloudMAC. However, CloudMAC has some significant benefits over CAPWAP:

State storage: in CAPWAP the state of a connection (association, authentication, encryption keys, MAC timers) is shared between controller and WTP. This makes the network unnecessarily complicated, as some state variables such as information associated STAs needs to be kept in sync. Hence WTPs cannot be exchanged easily during run-time. In CloudMAC though, all network state except for MAC timers and frames queued for transmission is kept in the VAP. The case study in section III-B demonstrated this benefit.

Simple deployment of new applications: there is no central platform to deploy applications on top of CAPWAP in an easy way. CAPWAP AP controllers are mostly proprietary systems that are hard to extend. In CloudMAC, new applications can be implemented on the OpenFlow controller. OpenFlow controllers such as NOX [20] allow implementing applications in high level programming or scripting languages.

Simplified administration: CloudMAC WTPs are simple devices. This reduces software complexity, chances for software bugs and allows for a slim hardware. Furthermore, as most functionality is provided by the VAP, adding support for new link layer encryption schemes, deploying new transmission power control etc. only requires updates only to the VAPs (there might be only one VAP for a whole network). This significantly decreases the effort required to update management software. CAPWAP WTPs are typically normal complex APs that support the CAPWAP protocol.

V. Summary and Outlook

Operating large WLAN deployments with heterogeneous hardware and software components provides a challenge to network operators. Enterprise WLAN management systems support the management of such networks, but are not flexible in implementing new services. In this paper, we have presented CloudMAC, a new WLAN management architecture, in which most processing and management functionality is concentrated on virtual access points that can be provided via existing cloud infrastructure. CloudMAC APs are simple devices that relay MAC frames between VAPs and mobile stations via an OpenFlow controlled network. Such OpenFlow based architecture provides many benefits for rapidly creating new services. We implemented such a new service on top of CloudMAC. The performance evaluation using our testbed has shown that CloudMAC achieves good performance and seamlessly interworks with standard IEEE 802.11 stations. As a future work, we plan to extend our case study to implement a full system for energy efficient operation of WLANs. In addition, we plan to evaluate the scalability of CloudMAC in terms of both number of user flows, mobile terminals and number of Virtual Access Points.

References

[1] S. Govindan, H. Cheng, Z. Yao, W. Zhou, and L. Yang, "Objectives for Control and Provisioning of Wireless Access Points (CAPWAP)," RFC 4564 (Informational), Internet Engineering Task Force, Jul. 2006. [Online]. Available: http://www.ietf.org/rfc/rfc4564.txt

[2] P. Calhoun, R. Suri, N. C. Winget, M. Williams, S. Hares, B. O'Hara, and S. Kelly, "Lightweight Access Point Protocol," RFC 5412 (Historic), Internet Engineering Task Force, Feb. 2010. [Online]. Available: http://www.ietf.org/rfc/rfc5412.txt

[3] "OpenWRT package repository," 2012. [Online]. Available: http://downloads.openwrt.org

[4] "Download Software - Cisco Systems," 2012. [Online]. Available: http://www.cisco.com/cisco/software/type.html?mdfid=281105734

[5] The OpenFlow Consortium, "Openflow switch specification 1.1," pp. 1–105, 2012. [Online]. Available: https://www.opennetworking.org/standards/intro-to-openflow

[6] U. Hoelzle, "Openflow @ google," Youtube online video, URL: http://www.youtube.com/watch?v=VLHJUfgxEO4, May 2012.

[7] 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*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 297–308.

[8] "IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1 –2793, 29 2012.

[9] B. Pfaff, J. Pettit, T. Koponen, K. Amidon, M. Casado, and S. Shenker, "Extending networking into the virtualization layer," in *Proc. of workshop on Hot Topics in Networks (HotNets-VIII)*, 2009.

[10] P. Dely and A. Kassler, "KAUMesh Demo," in *Proc. of 9th Scandinavian Workshop on Wireless Ad-hoc and Sensor Networks*, 2009.

[11] "Linux wireless wiki," 2012. [Online]. Available: http://http://linuxwireless.org/

[12] The OpenFlow Consortium, "Openflow website." [Online]. Available: http://www.openflowswitch.org/wp/downloads/

[13] N. Bayer, D. Sivchenko, H. Einsiedler, A. Roos, A. Uzun, S. Gondor, and A. Kupper, "Energy optimisation in heterogeneous multi-rat networks," in *Intelligence in Next Generation Networks (ICIN), 2011 15th International Conference on*, oct. 2011, pp. 139 –144.

[14] T. Hamaguchi, T. Komata, T. Nagai, and H. Shigeno, "A Framework of Better Deployment for WLAN Access Point Using Virtualization Technique," in *Proc. of IEEE WAINA 2010*, ser. WAINA '10, 2010, pp. 968–973.

[15] O. Braham and G. Pujolle, "Virtual wireless network urbanization," in *Proc. of NOF 2011*, Nov. 2011, pp. 31 –34.

[16] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A. Moore, A. Koliousis, and J. Sventek, "Control and understanding: Owning your home network," in *Proc. of COMSNETS 2012*, jan. 2012, pp. 1 –10.

[17] Y. Yiakoumis, K.-K. Yap, S. Katti, G. Parulkar, and N. McKeown, "Slicing home networks," in *Proc. of the 2nd ACM SIGCOMM workshop on Home networks*, ser. HomeNets '11, 2011, pp. 1–6.

[18] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for Wireless Mesh Networks," in *Proc. of IEEE International Workshop on Wireless Mesh and Ad Hoc Networks (WiMAN 2011)*, jul 2011.

[19] L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, and T. Vazao, "Towards programmable enterprise wlans with odin," in *Proceedings of the first workshop on Hot topics in software defined networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 49–54.

[20] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 105–110, July 2008.