ACM MobiCom 2012 Poster: CloudMAC - Towards Software Defined WLANs

Jonathan Vestin[a]
impclaw@gmail.com

Peter Dely[a]
peter.dely@kau.se

Andreas Kassler[a]
andreas.kassler@ieee.org

Nico Bayer[b]
nico.bayer@telekom.de

Hans Einsiedler[b]
hans.einsiedler@telekom.de

Christoph Peylo[b]
christoph.peylo@telekom.de

[a]Computer Science Department, Karlstad University, Karlstad, Sweden
[b]Telekom Innovation Laboratories, Berlin Germany

Traditional enterprise WLAN management systems are hard to extend and require powerful access points (APs). In this paper we introduce and evaluate CloudMAC, an architecture for enterprise WLANs in which MAC frames are generated and processed on virtual APs hosted in a datacenter. The APs only need to forward MAC frames. The APs and the servers are connected via an OpenFlow-enabled network, which allows to control where and how MAC frames are transmitted.

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. In addition, protocols such as CAPWAP [5] and LWAPP [2] allow vendor independent configuration of APs.

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

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 [10], as one instance SDN, has recently gained a lot of attention. For example, Google has replaced all its inter-datacenter backbone routers with simple OpenFlow-enabled devices, that are controlled from applications run on standard servers [8]. Thereby the operational expenditure can be decreased and network flexibility can be increased.

To get similar benefits in WLANs, we introduce CloudMAC, an alternative management architecture in which APs just for forward MAC frames. Other functions, such as processing MAC frames, are implemented on standard servers that are operated in data centers and can be provided via cloud computing infrastructure. The main contributions of this paper are an implementation of this architecture, a performance evaluation and a list of applications that are enabled through CloudMAC.

The remainder of the paper is organized as follows: In §II we present the architecture and implementation of CloudMAC. We evaluate the performance in §III and outline benefits and future applications in §IV. Related work is reviewed in §V. The paper is concluded in §VI.

II. Architecture and Implementation

II.A. Overview

Figure 1 shows a CloudMAC network, which consists of Virtual APs (VAPs), Wireless Termination Points (WTPs), an OpenFlow switch, an OpenFlow controller and tunnels to connect the entities.

WTPs are slim APs with WLAN cards operated in monitor mode (allows to send and receive raw MAC frames). The cards can send and receive MAC frames, but do not generate frames by themselves. Only ACK frames, which have very tight time constraints are generated by the cards. Each card includes a register of MAC addresses which it generates ACKs for. Other MAC frames are generated and received by VAPs.

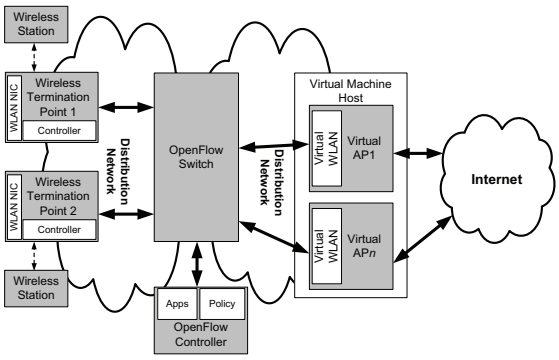

Figure 1: Architecture of a CloudMAC network

VAPs are operating system instances run on a virtualization host, such as Xen or VSphere Center. Each VAP has one or several virtual WLAN cards. A virtual WLAN card is driver, that appears to the operating system and user space applications like a normal physical WLAN card. All standard WLAN management tools can be used to set parameters of this virtual WLAN cards.

To send a packet from a VAP to a station, the virtual WLAN card adds a radio-tap and an IEEE 802.11 header and encrypts the frame (if WPA or WEP is used). The radio-tap header informs the physical card which modulation/coding scheme and transmission power to use. The frame is sent via layer 2 tunnels and an OpenFlow switch from the VAP to the physical card of a WTP. The WTP then transmits the frame to the station. In the opposite direction, from a station to the VAP, the process is done in the reverse order: the STA sends a frame, the WTP ACKs upon reception. The frame is forwarded to the VAP via a tunnel, where the virtual WLAN card receives it and hands it over to the networking stack of the operating system.

The OpenFlow switch and controller take a central role in the CloudMAC architecture. 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 a VAPs (more specifically, the virtual WLAN cards) and WTPs. Hence, by reconfiguring the switch table, one AP can easily be moved from one WTP to another. 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, in which multiple logical networks run on the same physical hardware.

VAPs run access point management software, for example to generate beacon frames, or respond to Association or Authentication MAC frames. As the virtual WLAN card appears like a real card, standard software such as hostapd can be used. 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.

II.B. Control Commands

Besides forwarding MAC frames, CloudMAC also allows a fine grained control of configuration commands. Configuration commands are used to configure the WLAN card and typically issued by a user space application in a virtual AP. For example, if an application requests the virtual WLAN card to change its channel, the request is intercepted in the virtual WLAN card driver. The driver then sends a configuration command packet to the OpenFlow switch. The switch forwards it to the OpenFlow controller. If the configuration command is permitted according to a user-configurable policy, the OpenFlow controller forwards the 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.

II.C. Implementation

We have implemented CloudMAC on the KAUMesh testbed [3]. The WTPs are Cambria GW2358-4 embedded devices and use OpenWRT Backfire. The VAPs are Debian 6.0 VMs on a VSphere Center installation. The VAPs run hostapd 0.6. We use OpenVSwitch 1.3.0 as OpenFlow switch, which runs in a VM on the same VSphere installation. Th switch controlled by custom made Python applications. The virtual WLAN card driver is based on the mac80211_hwsim driver, which was modified to allow MAC frame injection.

III. Performance Evaluation

Potential performance degradations may result from distributing MAC processing as done in CloudMAC. Hence, we compared the performance of CloudMAC with a normal WLAN AP (using the same hardware).

III.A. Microbenchmarks

With ping and iperf we measured the round trip time (RTT) and the throughput between a station and the (V)AP. Figure 2 shows the ECDF of the round trip time and the throughput for different TCP segment sizes. When CloudMAC is used instead of a simple

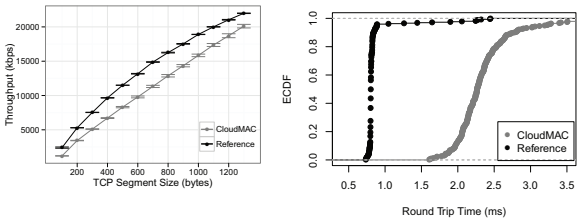

Figure 2: TCP throughput and Ping round trip time (error bars are standard deviation)

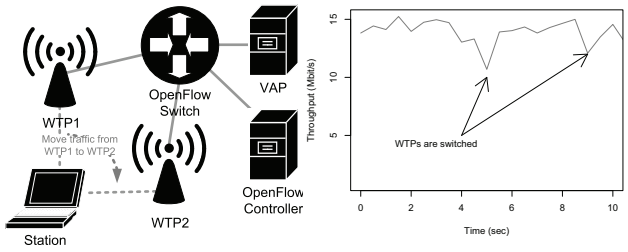

Figure 3: Scenario and throughput while switching WTPs (after 5 and 9 seconds)

AP, the mean RTT increases from 0.85 ms to 2.35 ms. This increase is due to the 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 (tested with Windows XP/Linux/MacOS X) to use CloudMAC VAPs.

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-level tunnels, which should result in improved performance.

III.B. Seamless AP Switch-Off

As demo application we implemented a seamless AP switch-off system. The idea behind that application is to save energy by switching of WTPs that cover the same area, if the network load is low. In traditional WLANs such a system would require stations to reassociate to a new AP, which leads a considerable service interruption. In CloudMAC the association state is kept in the VAP, so that moving traffic from one WTP to another only requires changes the forwarding rules in the OpenFlow switch and updating the register of MAC addresses that ACKs are generated for.

We evaluated if CloudMAC causes any noticable interruption when moving traffic from one WTP to another in the scenario depicted in Figure 3. The station downloads traffic from the VAP. After 5 and 9 seconds, the OpenFlow rules and MAC registers are changed, so that the traffic is moved from WTP1 to WTP2 and back again. As Figure 3 shows, there are small losses of throughput during moving. Moving traffic might cause some packets already buffered in the WLAN card to be lost or reordered. Nevertheless, the loss is so small that it is negligible in practice.

IV. CloudMAC Benefits and Future Applications

CloudMAC benefits compared to today's WLANs are:

Thin APs: CloudMAC WTPs only need a basic WLAN driver and a small control application. This reduces software complexity, chances for software bugs and allows for a slimmer hardware. Furthermore, WTPs do not keep association and authentication state information and can therefore easily be replaced during operation (see also §III.B).

Simplified administration: For example, to add support for a new link layer encryption scheme or remove bugs in an existing one, only the VAPs (which might be only one for a whole network) need to be updated. This significantly simplifies administration.

Simple deployment of new applications: Below we will outline several applications, that can be implemented hardware and vendor independent in high level programming language on an OpenFlow controller such as NOX [6].

Integration with OpenFlow networks: OpenFlow is likely to supersede many of today's network management protocols. CloudMAC is fully integrated into the OpenFlow architecture and can utilize OpenFlow infrastructure, such as switches and controller.

CloudMAC enables a range of new applications such as:

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 and thereby reduce the available capacity for data transfers. CloudMAC enables the following scenario: By default, the OpenFlow switch does not forward beacon frames from a VAP to the connected WTP. When a new user arrives and sends a probe requests (that sometimes includes the SSID of the desired network), an application on the OpenFlow controller inspects the probe request and dynamically en-

ables the beacon transmission. The new user now receives the beacon and can connect to the network. If the probe request includes no SSID, historical usage data and the user's MAC address can be used to identify the desired network. Thereby the number of beacons on the wireless medium can be reduced.

Downlink Scheduling: All traffic between the virtual APs and the WTPs pass 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.

Dynamic Spectrum Use: In CloudMAC, one virtual WLAN card can be connected to several physical WLAN cards. This allows the following application: one WTP contains several physical cards, operated with the same MAC address, but on different channels. The physical cards periodically monitor the channel utilization. If a station is currently using a channel with high external interference, the OpenFlow controller creates a IEEE802.11h action frame to instruct the station 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 virtual AP (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 (which is mandated by IEEE 802.11a/n).

V. Related work

Related work falls into three main areas:

Split MAC: With a split MAC, some MAC frames are generated in a central server, others on the local WLAN card. CAPWAP [5] and LWAPP [2] implement such a split MAC. In comparison to CloudMAC, they only generate frames for association and authentication, whereas CloudMAC generates all MAC frames, except for ACKs. This makes CloudMAC WTPs simpler and more flexible, as they do not contain any state information of associated STA.

AP virtualization: Previously, [7] and [1] proposed to run an OS hypervisor on APs and thereby provide virtual APs. However, such full AP virtualization requires powerful APs (often also x86-based APs), which are not typical. 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. This approach does not allow for fine-grained access control on the virtual WLAN cards (for example by different users).

OpenFlow wireless: OpenFlow has been applied in the context of wireless networks in [9], [11], [4]. Those works focus on IP-layer management and on authentication and QoS-provisioning through OpenFlow.

VI. Conclusions

We have presented a new WLAN management architecture called CloudMAC. CloudMAC implements an OpenFlow-controlled split MAC. The performance evaluation has shown that CloudMAC's performance is similar to normal WLANs and meets the timing requirements to interwork with standard IEEE 802.11 stations. CloudMAC enables a range of new applications, which we plan to implement in future.

References

[1] O. Braham and G. Pujolle. Virtual wireless network urbanization. In *Proc. of NOF 2011*, pages 31 –34, nov. 2011.

[2] P. Calhoun, R. Suri, N. Cam Winget, M. Williams, S. Hares, B. O'Hara, and S. Kelly. Lightweight Access Point Protocol. RFC 5412 (Historic), February 2010.

[3] Peter Dely and Andreas Kassler. KAUMesh Demo. In *Proc. of 9th Scandinavian Workshop on Wireless Ad-hoc & Sensor Networks*, 2009.

[4] Peter Dely, Andreas Kassler, and Nico Bayer. OpenFlow for Wireless Mesh Networks. In *Proc. of WiMAN 2011*, jul 2011.

[5] S. Govindan, H. Cheng, ZH. Yao, WH. Zhou, and L. Yang. Objectives for Control and Provisioning of Wireless Access Points (CAPWAP). RFC 4564 (Informational), July 2006.

[6] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38:105–110, July 2008.

[7] Tsuyoshi Hamaguchi, Takuya Komata, Takahiro Nagai, and Hiroshi Shigeno. A framework of better deployment for wlan access point using virtualization technique. In *Proc. of WAINA 2010*, pages 968–973, Washington, DC, USA, 2010.

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

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

[10] The OpenFlow Consortium. Openflow specification 1.1, 2012.

[11] Yiannis Yiakoumis, Kok-Kiong Yap, Sachin Katti, Guru Parulkar, and Nick McKeown. Slicing home networks. In *Proc. of HomeNets '11*, pages 1–6, 2011.