CloudMAC Installation Guide

Tong Li, Kun Yang

Any question please mail to lt@thucsnet.org

January 2015

Content

1 Introduction

Traditionally, Access Points (APs) in Wireless Local Area Networks (WLANs) were devices with low processing power and little intelligence. However, this design paradigm has gradually been abandoned and APs are getting more powerful and concentrate more functionality.

CloudMAC is a new management architecture for WLANs. The key idea of CloudMAC is to split up a WLAN AP into a physical AP, which just forwards MAC frames, and a virtual AP, which is hosted in a virtual machine in a data center or the cloud and contains all functionality such as MAC frame generation and authentication services. The virtual and the physical AP are connected via an OpenFlow enabled network.

1.1 System Requirements

To install CloudMAC, you need:

• Two physical PCs with at least 4 GB of RAM and two CPU cores. One machine will contain the VAP, another one will contain the OpenFlow switch (OpenVSwitch) and the CloudMAC controller (Here we have only one OpenFlow switch, the controller is setup locally).

• The machine containing VAP should have at least two network card, one (eth1) to connect to Internet, one (eth0) to connect to OpenFlow switch.

• One common switch(Cisco switch) to provide connection for three entities: the VAP machine, OpenVSwitch, and WTPs. Just a very traditional one is OK.

• Several domestic routers support OpenWrt OS to act as WTPs, here we use TP-LINK WDR3600.

Note that in this installation manual we suppose that software OpenVSwitch is used as OpenFlow switch. However, it is possible to use any (hardware) switch, that supports OpenFlow 1.1. and GRE tunnels.

1.2 Installation Overview

As the authors of the paper described, Figure 1 shows a system overview of CloudMAC. A CloudMAC installation consists of a Virtual Machine (VM) host, that contains VMs for the management of CloudMAC and Wireless Termination Points (WTPs) which forward MAC frames between the management machines and the user stations. WTP and VAP are connected to the OpenFlow switch using GRE Tunnel. However, in this installation manual we use one physical host (username: switch) to serve as Controller VM and Switch VM, we use another physical host (username: vap) to serve as VAP VM. Physical collectivity is detailed in Appendix Section 5.

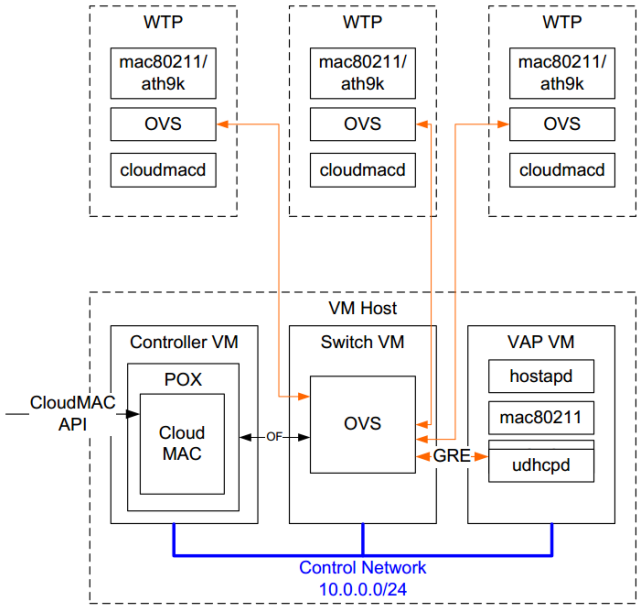

Figure 1: CloudMAC system overview

In more detail, the following components need to be installed and configured:

• Hosts: One machine install Ubuntu Linux 12.04 desktop for the VAP, one machine install Ubuntu Linux 12.04 desktop for the OpenFlow controller (POX) and the OpenFlow switch(OpenVSwitch 1.9).

• Virtual Access Points (VAPs): host with AP management tools
– hostapd: generates beacon frames, authentication etc., vanilla version
– mac80211: virtual WLAN card (CloudMAC modified version)
– udhcpd: dhcp server allocating IP address to OpenFlow switch, WTPs, and work stations.
– Configuration scripts

• Wireless Termination Points (WTPs): lightweight APs installed OpenWrt 12.09(attitude adjustment)

– mac80211/ath9k: modified to allow better control of injected frames
– cloudmacd: CloudMAC control daemon
– Configuration scripts

• Switch: connects the VAPs and the WTPs using GRE tunnels
– OpenVSwitch: OpenVSwitch version 1.9

• Controller: controls the OpenFlow switch
– POX Betta
– CloudMAC extension

In the following, we explain the various installation steps in order to get the whole CloudMAC up and running. However, it is advised to do this in the following order.

1. Install and start the VAPs

2. Install and start the OpenFlow Switch

3. Install and start the WTPs

2 Installation of VAP

2.1 Basic System

After install Ubuntu Linux desktop Version 12.04 (default installation). Install OpenSSH (for management purposes). In addition we install DHCP daemon. Then as our 802.11 driver can not run on Kernel Version 3.*, you need to revert back to Kernel Version 2.6.32. We do all this by:

- sudo apt-get install wireless-crda iw libnl-dev udhcpd
- wget http://ftp.halifax.rwth-aachen.de/ubuntu/pool/main/l/linux/linux-headers-2.6.32-22_2.6.32-22.36_all.deb
- wget http://security.ubuntu.com/ubuntu/pool/main/l/linux/linux-image-2.6.32-22-generic_2.6.32-22.36_amd64.deb
- wget http://security.ubuntu.com/ubuntu/pool/main/l/linux/linux-headers-2.6.32-22-generic_2.6.32-22.36_amd64.deb
- sudo dpkg -i linux-image-2.6.32-22-generic_2.6.32-22.36_amd64.deb
- sudo dpkg -i linux-headers-2.6.32-22_2.6.32-22.36_all.deb
- sudo dpkg -i linux-headers-2.6.32-22-generic_2.6.32-22.36_amd64.deb
- sudo vi /etc/default/udhcpd

  --> change to : DHCP_ENABLED="yes"

Update grub to make kernel version 2.6 the default boot image:

- sudo vi /etc/default/grub
  --> change to
  GRUB_ DEFAULT = " Advanced options for Ubuntuu > Ubuntu , with Linux 2.6.32-22-generic"
  Here we should change the number other than input the words, maybe from the default 0 to 2(see your system boot menu).
- sudo update-grub

After rebooting install OpenVSwitch:

- sudo apt-get install dkms openvswitch-common openvswitch-switch openvswitch-datapath-dkms

2.2 Installation of the Modified mac80211 Sub-System

CloudMAC brings along a modified version of the mac80211 subsystem in the Linux kernel for the machines where the VAP runs. You can download it from the cloudMAC code basin repository and compile it afterward:

- sudo apt-get install git build-essential
- mkdir git
- cd git
- git clone git://codebasin.net/cloudmac.git
- cd cloudmac/compat-wireless-2010-10-19
- make
- sudo make install

To verify that the compilation worked you can load the CloudMAC virtual WLAN driver

- sudo modprobe mac80211_hwsim
- dmesg

You should see a line "CloudMAC driver loaded".

2.3 Installation of hostapd

Run commands as follows:

- sudo apt-get install libssl-dev
- cd ~
- cd git/cloudmac/hostapd/hostapd-0.7.3/src
- make
- cd ..
- cd hostapd
- cp defconfig .config
- make
- sudo make install

You can check whether it works by

- hostapd –v

2.4 Configuration scripts of the VAP

Replace the following files for configuration. The file cloudmac-init.sh to be replaced is in "/cloudmac/vap-config/cloudmac-init.sh", and udhcpd.conf is in "/etc/udhcpd.conf".

-  cloudmac-init.sh  udhcpd.conf

We should modify the last two lines of udhcpd.conf on demand, in order to allocate fixed IP to WTP and OpenFlow switch:

static_lease MAC-of-WTP 10.0.0.10

static_lease MAC-of-OpenFlow-switch 10.0.0.3

The configuration script file cloudmac-init.sh determines various operational modes of the VAPs. Here are some details of the script.

cloudmac-init.sh script parameters
= =
NUMRADIOS=4

SWITCH_IP=10.0.0.3

SWITCH_KEY=10

CHANNEL=6

SSID="CLOUDMAC"

The parameters have the following meaning:

NUMRADIOS=4

determines the number of VAPs to create. Each VAP issues its own BSSID. In this case, 4 VAPs are created. The number of WTPs equals to the number of VAPs.

SWITCH_IP=10.0.0.3

determines the IP address of the OpenFlow Switch. In this case it is 10.0.0.3

SWITCH_KEY=10

defines the key for the GRE tunnel between the VAP host and the OpenFlow switch. In this case the key is 10. Note, that this value must match the parameter VAP_KEY in the config file of the OpenFlow switch.

CHANNEL=6

determines the channel that the radios of the WTPs are operating. In this case they use channel 6 in the 2.4 GHz band using 802.11g.

SSID="CLOUDMAC"

determines the SSID used by the WTPs. As in this example we create 4 VAPs, the following SSIDs are broadcaste : CLOUDMAC0, CLOUDMAC1, CLOUDMAC2, and CLOUDMAC3.

In addition, the script gives you the possibility to influence various parameters of the wireless card behavior. To do this, you have to modify in the script function *create_hostapdconf()* various parameters on demand, such as add encryption assess for WTPs, WPA1 and WPA2 are both OK.

2.5 Start of VAP

Run cloudmac-init.sh and Run nat.sh to enable router feature in vap.

- sudo sh cloudmac-init.sh
- sudo sh nat.sh

Run following commands to enable two default gateways on vap

- echo “1 admin” >> /etc/iproute2/rt_tables (This command should be run only once)
- sudo sh ip_rule_config.sh

If dhcp server is not running on VAP or you want to see whether new IP address is allocated, run the commands below:

- sudo /etc/init.d/udhcpd start
- sudo udhcpd –fS /etc/udhcpd.conf

Every time we start VAP, just run the red commands.

3 Installation of OpenFlow Switch

3.1 Basic System

After install Ubuntu Linux desktop Version 12.04 (default installation). Install OpenSSH (for management purposes). Run commands below to install Openvswitch1.9:

- sudo apt-get install git dkms debhelper autoconf libssl-dev python-all linux-headers-generic pkg-config python-qt4 python-twisted-conch
- wget http://openvswitch.org/releases/openvswitch-1.9.0.tar.gz
- tar xf openvswitch-1.9.0.tar.gz
- cd openvswitch-1.9.0
- dpkg-buildpackage -us -uc -b
- cd ..
- sudo dpkg -i openvswitch-common_1.9.0-1_amd64.deb openvswitch-datapath-dkms_1.9.0-1_all.deb openvswitch-switch_1.9.0-1_amd64.deb
- /etc/init.d/openvswitch-switch start

Now we need to get the configuration scripts for CLOUDMACs OpenFlow Switch. In order to do that we need to get the resource files:

- git clone git://codebasin.net/cloudmac.git

3.2 Configuration scripts of the OpenFlow Switch

Replace cloudmac.cfg in /cloudmac/pox-master with attached file below, we should modify it if we add more WTPs:

- 
cloudmac.cfg

Replace switch-config.sh in /cloudmac/switch-config with below attached file

- 
switch-config.sh

The configuration script file switch-config.sh determines various operational modes of the OpenFlow Switch. Here are some details of the script.

switch-config.sh script parameters
= =
CONTROLLER=127.0.0.1:6633

VAP_IP=10.0.0.1

VAP_KEY=10

WTP_PREFIX=10.0.0

WTPS="10 11 12 13"

The parameters have the following meaning:

CONTROLLER=127.0.0.1:6633

defines the IP-address and port of the POX, the openflow controller. In this example, it runs on the same machine (127.0.0.1) as the Op enFlow Switch listening on its default p ort 6633.

VAP_IP=10.0.0.1

defines the IP address of the VAP. In this example, VAP run in the machine with the IP address 10.0.0.1.

VAP_KEY=10

defines the KEY of the GRE tunnel between the OpenFlow switch and the VAPs. In this example, the key is 10.

WTP_PREFIX=10.0.0

defines the IP prefix of the WTPs. In this example, the prefix is 10.0.0

WTPS="10 11 12 13"

defines the suffix of the supported WTPs. In this example, we have in total 4 WTPs. The WTP addresses are thus 10.0.0.10, 10.0.0.11, 10.0.0.12 and 10.0.0.13. Note that you have to install the WTP software on those machines (such as TP-LINK Router).

3.3 Start of OpenFlow Switch

In order to get the OpenFlow Switch operational, you need to get first the switch up and running and then you need to start the controller Pox. You start the switch by start

- sudo sh switch-config.sh

Then you start the Pox by starting the Pox startup script start-pox.sh in the sub-directory pox-master.

- sudo sh ../pox-master/start-pox.sh

In case you need to configure the Pox differently, you can change the address where the Pox is listening and the port accordingly by changing the start-pox.sh script.

Note that POX does not support OpenFlow 1.3, it can only support OpenFlow1.0. If error occurred please set the OpenFlow version manually after running the switch-config.sh:

- sudo ovs-vsctl set bridge br0 protocols=OpenFlow10

Every time we start OpenFlow switch, just run the red commands.

4 Installation of WTP

4.1 Basic System

We compile Openvswitch 1.9.0 with OpenWrt 12.09 attitude adjustment. More openwrt version please refer to https://dev.openwrt.org/wiki/GetSource. Now prepare pre-compile environment in Ubuntu:

- sudo apt-get install build-essential subversion git-core libncurses5-dev zlib1g-dev gawk flex quilt libssl-dev xsltproc libxml-parser-perl unzip

Get source of openwrt (12.09 attitude adjustment) and config it:

- svn co svn://svn.openwrt.org/openwrt/branches/attitude_adjustment
- cd $TOPDIR
- echo 'src-git openvswitch git://github.com/schuza/openvswitch.git' >> feeds.conf
- Run ./scripts/feeds update -a
- Run ./scripts/feeds install -a
- make menuconfig
- select Network -> openvswitch-switch, openvswitch-brcompat and openvswitch-controller

Other selections such as LUCI please refer to the internet…

- echo '# CONFIG_KERNEL_BRIDGE is not set' >> .config

We compile openwrt after including patch for code changes in mac layer (this patch is based on attitude adjustment 12.09, in "..\wtp-config\950-cloudmac_add_code.patch")

- 
950-cloudmac_add_code.patch

Place the patch file in ../package/mac80211/patches. Then

- cd $TOPDIR
- make

If patch file error occurred, then try the patch file in " ..\wtp-config\OpenWrt1407-instruction\950-cloudmac_add_code.patch" or fix it all by yourself. After a long compile time, we get the firmware (*.bin). Find the right one to flash it into TPLINK WDR3600 and configure it (use LUCI or Telnet tool to set up login password, etc.).

WTP needs complete python package to support, if the ROM of WTP is big enough(for example, boot system from USB device, see another guide file "wtpcompile.docx"), we can just install Python as follows:

- opkg update
- opkg install python

However, if we do not have enough space to install Python to ROM(use command "df –h" to check), we can just install Python to RAM (this may be not stable):

- opkg update
- opkg install python –d ram
- export PATH=$PATH:/tmp/usr/bin

4.2 Start of WTP

Now we need to get the configuration scripts for WTPs. In order to do that we need to download the original files to our computer(ubuntu),

- git clone git://codebasin.net/cloudmac.git

Here we only need the file folder "cloudmacd", use scp tool to copy the file folder "cloudmacd" from ubuntu to WTP's directory "/", copy the configuration file wtp-config.sh attached below (here we install Python in RAM) to "/root/". For example,

- scp cloudmacd.tar.gz root@192.168.1.1:/
- scp wtp-config.sh root@192.168.1.1:/root
- 
  wtp-config.sh

Set DHCP mode for br-lan interface of the router(you can do it by LUCI), so that the DHCP server on VAP will be able to allocate the IP address 10.0.0.10 to the router(as configured in /etc/udhcpd.conf).

This file wtp-config.sh attached is in "..\wtp-config\wtp-config.sh", if you boot from USB device and install Python in ROM, then use the file in " ..\wtp-config\OpenWrt1407-instruction\wtp-config.sh" as the command path of Python is different .

Run the shell script to start WTP, this script will call the python code in the folder of "cloudmacd".

- sh wtp-config.sh

As we have only setup one WTP in our scripts, with these settings above, WTP should able to broadcast CLOUDMAC0 SSID.

Every time we start WTP, just run the red command.

5 Appendix

5.1 IP addresses

Machine	Interface	IP address
VAP(connect to Internet)	eth1	192.168.2.* gateway:192.168.2.253
VAP	br1(virtual)	10.0.0.1
VAP	eth0	Bridged with br1 with no IP address
Cisco switch	Not necessary	Not necessary
Pox controller (OpenFlow switch)	eth0	10.0.0.3
WTP	br-lan	10.0.0.10

5.2 Passwords

Machine	User	Password
VAP	vap	essexuser
Cisco switch	cisco	cisco
Pox controller (OpenFlow switch)	switch	essexuser
WTP	root	toor

5.3 Physical Collectivity