November 18, 2012

- by Lasse Christiansen
- in Linux, RPi
- Comments (28)

Raspberry Pi Tutorial – Connect to WiFi or Create An Encrypted DHCP Enabled Ad-hoc Network as Fallback

In this post I describe how I have configured my Raspberry Pi (RPi) to first attempt to connect to WiFi and if that fails, create and use an ad-hoc network as fallback (in this way I can always reach the RPi via SSH). The blog post is based on the following "How To" from the Raspberry Pi forum: http://www.raspberrypi.org/phpBB3/viewtopic.php?t=19517&p=190855 – however, I have introduced a level of more detail and a couple of modifications in order to get faster boot time and support for multiple wireless networks (see my previous RPi blogt post suvery for details on which parts of that "How To" I think are good as well as which I think can be improved).

Hardware

- The WiFi adapter I have used for this tutorial is the *Edimax EW-7811Un* (more details here) which seems to be one of the more popular WiFi adapters for the RPi (at least based on the amount of forum posts where it appears).

Software

This tutorial has been tested with success on:

- 2012-09-18-wheezy-raspbian
- 2012-10-28-wheezy-raspbian

What We Are Going To Do

To reach our goal, I will walk through the following 7 steps:

1. Getting Wireless Interface Name and Hardware Address
2. Connect to WiFi
3. Install and Configure DHCP Server
4. Update `interfaces` Config File
5. RPi Network Conf Bootstrapper
6. Prevent DHCP From Starting At Boot
7. Reboot and Test

So basically, the main idea is that we use `wpa_supplicant` for maintaining a connection to WiFi, and we use an ad-hoc network as fallback if we cannot connect to WiFi on boot. Further, to make it easier to connect and communicate with the RPi via the ad-hoc network, we setup a DHCP server for that network as well.

Step 1 – Getting Wireless Interface Name and Hardware Address
In the coming steps we are going to use the name and hardware address (MAC address) of your wireless device/interface. To see that you can use:

```
1 ifconfig
```

… or:

```
iwconfig
```

The name of my wireless interface is: `wlan0`

… and the pattern of its hardware address is: `00:aa:11:bb:22:cc`

Step 2 – Connect to WiFi

First of all, use `wpa_cli` or the "WiFi Config" GUI in lxde to setup any WiFi connections on the RPi.

The GUI is self-explanatory, but the command line approach can be a bit tricky if you don't know where to start. Based on:

- http://sirlagz.net/2012/08/27/how-to-use-wpa_cli-to-connect-to-a-wireless-network/
- http://superuser.com/a/341440/127795

… the steps are:

Open the command line interface (cli) for `wpa_supplicant`:

```
wpa_cli
```

Make it scan for networks:

```
> scan
```

List the results:

```
> scan_results
```

From the list of networks, you should be able to see the SSID of the network you are going to connect to. If that is the case, we are now going to add that network to `wpa_supplicant`. Make sure to note the number returned by the `add_network` command as that identifies the new network (if you forget, you can use `list_networks` to view the list of networks known to `wpa_supplicant`):

```
> add_network
0
```

Then we need to tell about the SSID and the WPA/WPA2 key of the network:

```
> set_network 0 ssid "mySSID"
> set_network 0 psk "myKey"
```

I have seen some guides saying that at this point, `wpa_supplicant` should automatically start connecting to the specified network, however, that is not what I have experienced. So, if `wpa_supplicant` does not automatically start to connect after pressing the commands above, use the following command to "force" it to connect:

```
> enable_network 0
```

Then, in order to persist the settings, we type:

```
> save_config
```

And finally, we quit the `wpa_cli`:

```
> quit
```

At this point, you should be connected to the desired WiFi.

If not, you might need to request an IP-address from the DHCP server on the WiFi network using:

```
sudo dhclient -1 wlan0
```

Step 3 – Install and Configure DHCP Server

Next, we are going to install the `isc-dhcp-server` DHCP server using `apt`:

```
sudo apt-get update
sudo apt-get install isc-dhcp-server
```

The installation itself should go smoothly, however, when the DHCP server is started

automatically after the installation it will probably report a failure saying that it is *"Not configured to listen on any interface"* - which makes sense as we haven't configured it yet. This error message is seen by taking a look at `/var/log/syslog` btw.
When the installation is done, we need to configure some defaults for the DHCP server for when it is started:

```
sudo nano /etc/default/isc-dhcp-server
```

The only change we need to do in this file is to add the name of our wireless interface to the `INTERFACES` setting, like:

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
```

In this way, we specify that we want the DHCP server to serve DHCP requests on the interface: `wlan0`
Next up is the config file for the DHCP server:

```
sudo nano /etc/dhcp/dhcpd.conf
```

My config file looks like this – see the added comments for details:

```
DHCPDARGS=wlan0; #args for the dhcpd daemon -> limit DHCP to the wlan0 inter
default-lease-time 600;
max-lease-time 7200;

option subnet-mask 255.255.255.0;
option broadcast-address 10.0.0.255;
option domain-name "RPi-network";
option routers 10.0.0.1; #default gateway

subnet 10.0.0.0 netmask 255.255.255.0 {
    range 10.0.0.2 10.0.0.20; #IP range to offer
}

#static IP-assignment
host myLaptop {
    hardware ethernet 11:aa:22:bb:33:cc;
    fixed-address 10.0.0.100;
}
```

Step 4 – Update `interfaces` Config File

In this step we are going to configure the network interface configuration information:

```
sudo nano /etc/network/interfaces
```

My interfaces file looks like this (see the added comments for details):

```
# start interfaces upon start of the system
auto lo wlan0

# register loopback interface
iface lo inet loopback

# use dhcp and allow interface to be started when kernel detects a hotplug e
allow-hotplug eth0
iface eth0 inet dhcp

# use manual ip configuration for wlan0 interface and allow hotplug as well
allow-hotplug wlan0
iface wlan0 inet manual
```

Step 5 – RPi Network Conf Bootstrapper

In this step we are going to do a bit of bash scripting in order to the tell RPi to create an ad-hoc network if it cannot connect and obtain an IP-address from one of our already known WiFi networks.

We want this code to be executed when the RPi has booted and "is ready for use", so we use

`rc.local` as the container of our script:

```
sudo nano /etc/rc.local
```

The script I provide below requires that you use `bash` as the interpreter for the `rc.local` file – to do this, you need to change the very first line (the shebang) to:

```
#!/bin/bash
```

The script I provide here (RPi Network Conf Bootstrapper) should be put at the end of the `rc.local` file – the script goes as follows:

```
# RPi Network Conf Bootstrapper

createAdHocNetwork(){
    echo "Creating ad-hoc network"
    ifconfig wlan0 down
    iwconfig wlan0 mode ad-hoc
    iwconfig wlan0 key aaaaa11111 #WEP key
    iwconfig wlan0 essid RPi      #SSID
    ifconfig wlan0 10.0.0.200 netmask 255.255.255.0 up
    /usr/sbin/dhcpd wlan0
    echo "Ad-hoc network created"
}

echo "================================="
echo "RPi Network Conf Bootstrapper 0.1"
echo "================================="
echo "Scanning for known WiFi networks"
ssids=( 'MyWlan' 'MyOtherWlan' )
connected=false
for ssid in "${ssids[@]}"
do
    if iwlist wlan0 scan | grep $ssid > /dev/null
    then
        echo "First WiFi in range has SSID:" $ssid
        echo "Starting supplicant for WPA/WPA2"
        wpa_supplicant -B -i wlan0 -c /etc/wpa_supplicant/wpa_supplicant.con
        echo "Obtaining IP from DHCP"
        if dhclient -1 wlan0
        then
            echo "Connected to WiFi"
            connected=true
            break
        else
            echo "DHCP server did not respond with an IP lease (DHCPOFFER)"
            wpa_cli terminate
            break
        fi
    else
        echo "Not in range, WiFi with SSID:" $ssid
    fi
done

if ! $connected; then
    createAdHocNetwork
fi

exit 0
```

Step 6 – Prevent DHCP From Starting At Boot

As we only want it to start if an ad-hoc network has been created, we need to tell the DHCP server not to start on boot:

```
sudo update-rc.d -f isc-dhcp-server remove
```

Step 7 – Reboot and Test

Now, everything should be setup and ready to work – so reboot the RPi and pay attention while it boots to make sure everything looks right.

```
sudo reboot
```

Finally, try to connect to the same network as the RPi and see if it responds to ping requests. Also, try to connect to its ad-hoc network and verify that the DHCP server is leasing you an IP.

Dones

If everything looked okay, and your RPi is either connected to your WiFi or emits and ad-hoc network, everything should be working as expected 😃 Otherwise, if you are experiencing problems, walk through the tutorial again and make sure that all your config files matches what is written in the tutorial – if that do not solve your problem, feel free to send me an e-mail or ask further question at the Raspberry Pi forum.

« Getting Started with My Raspberry Pi – Part 2: RPi Specs and Peripherals
Pro tip: Show "missing" Apps in Windows Store »

28 Responses to "Raspberry Pi Tutorial – Connect to WiFi or Create An Encrypted DHCP Enabled Ad-hoc Network as Fallback"

1. 

   Chris Hansen January 5, 2013 at 10:41 pm #

   I'm getting a syntax error on ssids=('MyWlan' 'MyOtherWlan') Any clue what that could be about?

   Reply

   - 

     Lasse Christiansen January 5, 2013 at 10:46 pm #

     Have you remembered to change the script interpreter to bash? (That is, changing the very first line of the rc.local script to: "# /bin/bash")

     Reply

     - 

       Chris Hansen January 5, 2013 at 11:08 pm #

       Just double checked, and I had a 2nd SD card laying around so I ran through the instructions again on a vanilla Wheezy image. Set for bash and still the same error on boot.
       Right after the echo for Scanning for known Wifi networks.
       /etc/rc.local: 32: /etc/rc.local: Syntax error: "C" unexpected

       Reply

       - 

Chris Hansen January 5, 2013 at 11:13 pm #

Correction: /etc/rc.local: 32: /etc/rc.local: Syntax error: "(" unexpected

Reply

Lasse Christiansen January 5, 2013 at 11:17 pm #

Oh, my bad – I've just realized that I have included the wrong syntax of how to specify the interpret Instead of "# /bin/bash" try this: "#!/bin/bash" – I was missing the exclamation mark (this has been in the blog post as well). Try it, and let me know if that made any difference

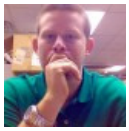
Chris Hansen January 5, 2013 at 11:23 pm #

I missed that twice >.< Works perfectly now.

Reply

Lasse Christiansen January 5, 2013 at 11:26 pm #

Great! Thanks for stopping by and letting me know

2. 

Lucas January 11, 2013 at 1:44 am #

I am getting an error on boot up and down saying that there is something wrong with /etc/network /interfaces.
I deleted everything in that file and pasted in your code. Is that what I was supposed to do?

Reply

Lucas January 11, 2013 at 1:52 am #

Here is the error I get. Any help is appreciated.

[....] Configuring network interfaces…/etc/network/interfaces:3: misplaced option ifup: couldn't read interfaces file "/etc/network/interfaces"
failed.

Reply

- 

  Lasse Christiansen January 11, 2013 at 7:29 am #

  Hi Lucas, thanks for your comment.

  I will double check the correctness of my interfaces file when I get my RPi booted later today.

  However, I briefly searched for this issue, and it appears to me that the typical main root cause of this, is a syntax error in the interfaces file. So, first of all, are you absolutely, 100% sure, that the file you are using are identical to the one I show in this post?

  If so, you could try again with an empty interfaces file and instead of copy pasting, try writing the exact content in the file – I have sometimes seen that doing a copy paste from one platform to the other induces some unwanted side-effects.

  Reply

  - 

    Lasse Christiansen January 11, 2013 at 8:07 pm #

    Hi Lucas, I have now compared the provided interfaces file in this blog post with the one currently runnin on my RPi – and they are identical. I also replaced my interfaces file on the RPi with a copy paste of the o from my blog post, and the interfaces settings loads without any issue on boot. So ye, currently, my only suggestion is that you start verifying that the interfaces file in this post is 100% equal to the one you use your RPi 😃 Let me know how it turns out 😃 Thanks.

    Reply

3. 

4. Owen January 17, 2013 at 10:22 pm #

I am so happy that this works.

First thing i did with my pi was make it an airplay receiver.

Second thing was this!

I now have a standalone device that i can plug into my Aux in my car or work and play music!!

Reply

5. Dan January 20, 2013 at 3:52 pm #

Thanks for the tutorial! Needed this for a project where the pi is in my car and needs to pull the latest code on boot if possible. Thanks a million

Reply

6. Michael King February 5, 2013 at 12:12 am #

First of all thank you for these detailed instructions. I am having an issue where ad-hoc wifi network is no longer available when the device connected to the ad-hoc network disconnects. This seems like perhaps there is another setting I need to make to tell the ad-hoc network to persist when a device disconnects?

When this happens, I have to reboot the Raspberry Pi which allows me to re-establish the ad-hoc connection. Note if I have the Raspberry Pi connected to an access point, the connection to the Raspberry Pi can always be re-established.

So it is as if the networking stack has crashed when in ad-hoc mode and a device disconnects.

Any advice here to help resolve this would be greatly appreciated.

Reply

7. Matt February 5, 2013 at 1:51 am #

Thanks for a great tutorial. Very useful. It's worked fine for me up until the part when i try to connect other devices to my ad-hoc network. They just time out when try to lease an ip address. If I allow my android phone (for instance) to use it's own ip settings, it finds an address (albeit outside the range supposed to be leased by the dchp), but i can't connect to the default gateway.

Any suggestions!? Many thanks!

Reply

Lasse Christiansen February 6, 2013 at 8:54 am #

Hi Matt and Michael, thanks for your comments – I will take a look at both during the weekend

Reply

Adrian Cristian Pop February 8, 2013 at 2:38 pm #

Thanks for the effort put into making this tutorial. I've done a little bit more and added a router script with iptables rules in case it makes a hotspot.

Reply

Lasse Christiansen February 8, 2013 at 2:42 pm #

Hi Adrian, that sounds great! Just out of interest, could you please share a link to a blog post describing your script (if a such exist)?

Reply

9. 

Lasse Christiansen February 10, 2013 at 10:53 pm #

Hi Matt and Michael, I have unfortunately been fully occupied the entire weekend, but if I get some more spare time in the next week, both of your "issues" are on my list and I will look into it.

Reply

10. 

Warren February 14, 2013 at 5:43 pm #

I followed the tutorial but and I was observing erratic behavior when the rc.local script creates the ad-hoc network and executes the 'iwconfig wlan0 mode ad-hoc' command. I was getting:
Error for wireless request "Set Mode" (8B06):
SET failed on device wlan0 ; Device or resource busy.

The strange thing was that this would occur only once every 2 or 3 times upon full (power-unplugged) reboot. I was using a Tenda Wi-Fi dongle plugged into a powered usb hub, but found that using a BT-Micro3H2X (combination Bluetooth WiFi) dongle plugged directly into the RPi, the problem has not reappeared.

Excellent tutorial; many thanks!

Reply

11. 

Lasse Christiansen February 17, 2013 at 11:31 am #

@Michael – I just tried connecting to my RPi, then disconnect and re-connect again – thus, I cannot reproduce the exact same problem. Which type of WiFi device are you using?

Reply

Lasse Christiansen February 17, 2013 at 11:36 am #

@Matt – have you tried connecting using a PC? As far as I know, Android devices are still not able to connect to Ad Hoc networks out of the box.

Reply

Matt February 25, 2013 at 7:55 pm #

Hi Lasse, thanks for your reply. Yes, I have tried it from a PC also, but I get a similar problem. The PC says that it is connected, but it is not leased an IP address, and does not receive any packets from the RPi. I've followed your tutorial exactly….very odd!

Reply

Lasse Christiansen February 25, 2013 at 9:10 pm #

Hi Matt, thanks for your reply I have a couple of questions that might help narrow the issue down – have you checked that the DHCP server is actually running? Is the DHCP server complaining about an invalid config file when started? Which WiFi device are you using? Have you tried starting the "RPi Network Conf Bootstrapper" from the terminal and checked that everything went well (that no errors were returned?)

Reply

13. 

Hans March 3, 2013 at 12:19 pm #

Hi Lasse, first i want to say is THANK YOU . I have RPi (rasbian wheezy, but 256MB version) and same wifi usb adaptor Edimax EW-7811Un. I would like create adhoc "server" on windows 7 (in this post – http://raspberrypi.org/phpBB3/viewtopic.php?f=63&t=34895) but i was unsuccessful. So i found your really great tutorial and do this way ("server" on raspberry) and this works. But when i start (boot) RPi, it stay while on "Obtaining IP from DHCP" then write on screen " DHCP server did not respond with an IP lease (DHCPOFFER) " and then "Failed to connect to wpa_supplicant – wpa_ctrl_open: No such file or directory " and finaly Creating ad-hoc network. This problem is now OK for me, because i need only Ad-Hoc network (for internet (for update) i use cable). Is there any way to choose what i want. I think in script will be something like this – echo "Choose: For wifi write "w" and for AdHoc write "a" " then user write W or A for option that he want. But that is just detail. More main thing i would like to change is transfer speed, edimax can work in 802.11 N mode, but when i wrote " iwconfig wlan0 " it show me only 54mb/s (so it is 802.11 G mode?). Is there any way to change transfer speed, because i stream video from USB camera with MJPG-STREAMER (by BobTech) and there is 3 seconds delay between reality and video, so i think that higher transfer speed is better.

THANK YOU AGAIN for this tutorial
Hans

Reply

Lasse Christiansen March 3, 2013 at 10:10 pm #

Hi Hans, thanks for your response – I'm glad the post have helped several RPi users out there Regarding your questions – I think a simple way of solving the "Failed to connect to wpa_supplicant – wpa_ctrl_open: No such file or directory" error is to simply call "createAdHocNetwork" directly in the bootstrapper and skip the bootstrapper's attempt to connect to an existing wireless network (which I understand is okay, since your are using cable for Internet access). In regards to the transfer rate of the network connection, I have seen a couple of blog posts mentioning that WEP is not supported on the N standard, meaning that if WEP is enabled, the device will automatically fall back to the g standard. So, you could try editing the script such that it do not set a WEP key, then connect and see what speeds are reported? Further, I have seen that it should be possible to "force" the device to use a certain standard by using the "rate" option for iwconfig – so that would also be an option to look further into

Reply

Trackbacks/Pingbacks

1. Raspberry Pi Prank Tutorial – Making Your Co-Worker's Desk "Magically" Go Up and Down | Lasse Christiansen Development - December 6, 2012

   [...] I have covered this topic in one of my previous blog posts, here: Raspberry Pi Tutorial – Connect to WiFi or Create An Encrypted DHCP Enabled Ad-hoc Network as Fall…. [...]

2. api.danielvagg.com – functional(ish) | Snapshots of Thoughts - January 26, 2013

   [...] So, I have written a wee chunk of code in '/etc/rc.local' using advice from this blog here! The endpoints are served using a simple enough Python service which accepts POST and GET. The [...]

Leave a Reply

Name

Email

Website

Your Comment

Submit Comment

Lasse Christiansen | @sw_lasse | lasse@lcdev.dk | ICQ# 150152933