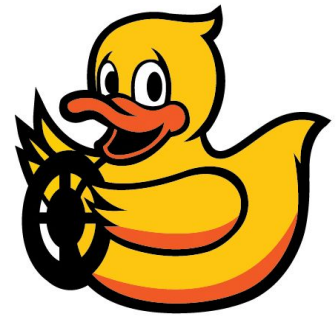


This document is the new version of [this](#).

Setup Step 1.7_v2

Setting up the Buffalo access point



[Step 1: Obtain unopened Buffalo](#)

[Step 2: Unboxing](#)

[Step 3: Warning sign](#)

[Step 4: labeling](#)

[Step 6: Connection](#)

[Step 7: Firmware update](#)

[Step 8: Execute "Local network configuration"](#)

[Step 9: Execute "Internet connection configuration"](#)

[Step 10: Verifying the network configuration](#)

[Step 11: Configuring the laptop](#)

[Step 11a: Configure Ubuntu native](#)

[Step 11b: Configure Ubuntu/Virtualbox](#)

[Step 11c: Configure Ubuntu/VMWare](#)

Step 0: Remove the Canakit or any other wifi adapter from the Pi

Step 1: Obtain unopened Buffalo



Remember to add “1” to the corresponding column [in this lab document](#).

Step 2: Unboxing

Be careful and do not destroy the box (we might return these).

Separate this stuff and keep it safe in a Duckiebox in 226:



These are the parts that the students will get:



Remember to add “1” to the corresponding column [in this lab document](#).

Step 3: Warning sign

The Raspberry Pi will be damaged by the Buffalo. Therefore, we put a warning sign.

Simplest way:



Then attach with clear tape:



Remember to add "1" to the corresponding column [in this lab document](#).

Step 4: labeling

We assign each Buffalo to a robot and we record the important data in the Scuderia document.

Go to the [Scuderia](#) and find the first student robot without the Buffalo columns filled.

Trying to connect. To edit offline
Put the information in the columns.

	A	B	AA	AB	AC
1			Buffalo		
2					
	Hostname	owner/maintainer	Name of initial Buffalo network	Buffalo password	Buffalo MAC
3					
4					
5	mercedes	Censi	Buffalo-A-7582	27833193	cce1d5307582

This information (network name, password, MAC) is written on the back of the Buffalo. The card that comes with it contains the name of the network and password but not the MAC.

Then put a label on the Buffalo.

The label should say exactly this:

```
<robot>
SSID: <robot>-5
PWD: "quackquack"
```

This is what it looks like.

Remember to add "1" to the corresponding column in this lab document.

Step 5: Download firmware:

We need firmware 1.3. The bin is in duckietown-data:

https://www.dropbox.com/s/r1s3xirhy36vo47/wmr433us_130.bin?dl=1

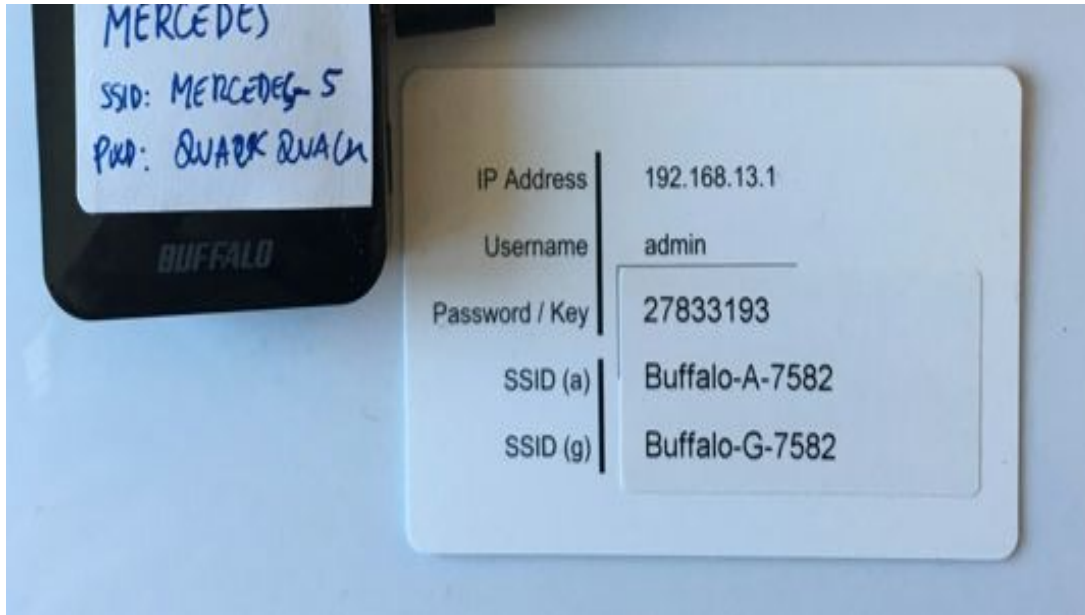
Finish the download before continuing to the next step.

Remember to add "1" to the corresponding column in this lab document.

Step 6: Connection

Choice:

- if your computer has 5ghz wifi, make sure the switch on the side of the buffalo is on "5 GHz". It is called duckietown-5. Otherwise make sure that your network connects to duckietown
- otherwise set it to 2.4 Ghz



Plug the **usb** cable of the Buffalo into the **battery** (not the PI - see warning sign).

Make sure the lights are lit.

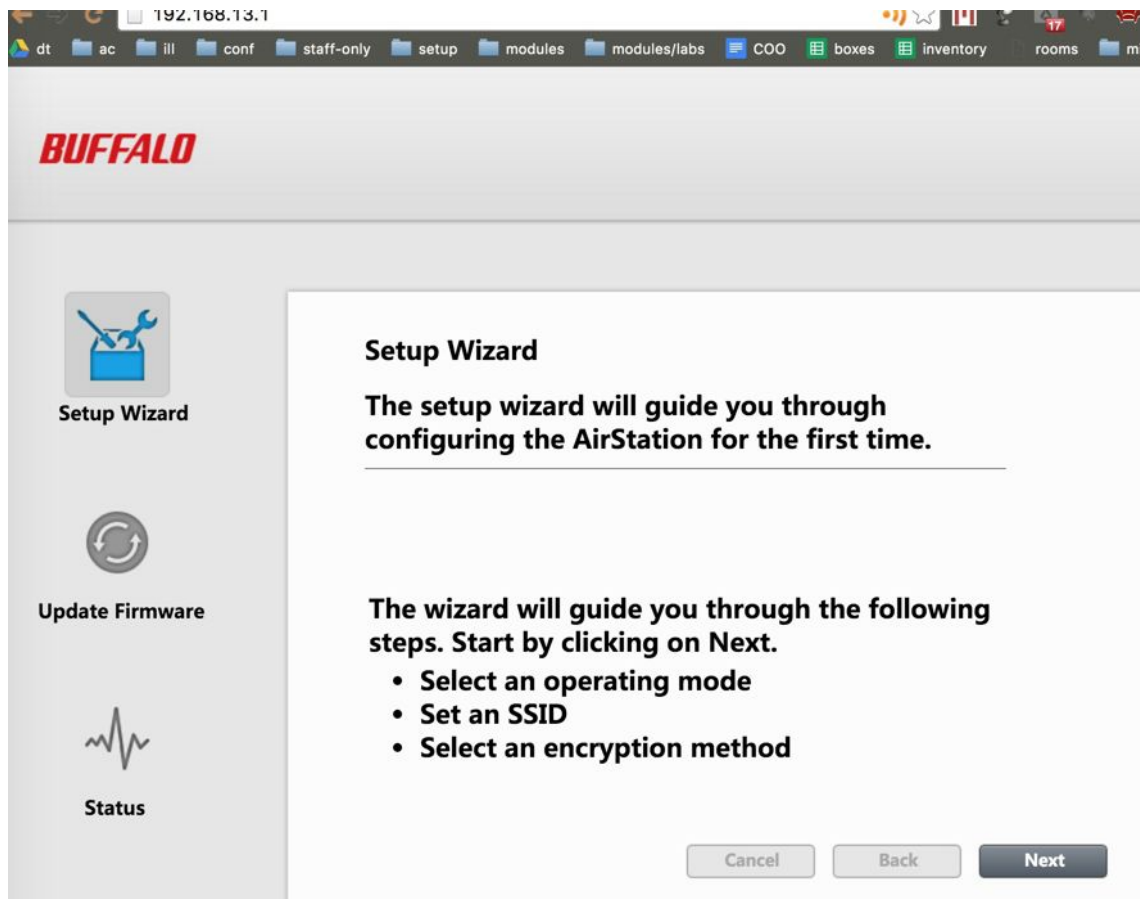


Plug one end of the short **ethernet** cable to the Buffalo, and the other to the Pi.

Now use your laptop to connect to the SSID that you noted before (in Scuderia). It will ask you for a password.

Once connected, connect to 192.168.13.1. Use user “admin” and password as before.

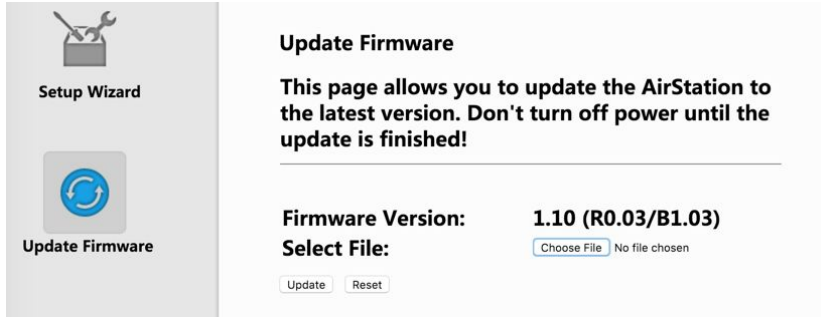
You will see something like this:



Remember to add “1” to the corresponding column [in this lab document](#).

Step 7: Firmware update

We will now use the firmware we downloaded in step 5 to update the Buffalo.

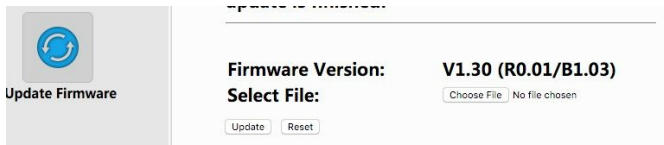


Upload the file and wait:



Reconnect to the Buffalo network and go to the same web page.

It should say:



Remember to add “1” to the corresponding column [in this lab document](#).

Step 8: Execute “Local network configuration”

Now go to Setup wizard (still at 192.168.13.1)

Click **next**.

Select **Local Connection**:

Select an Operating Mode

Wired Connection: Select this if a wired network is available.
 - Internet: Wired
 - LAN: Wireless

Wireless Connection: Select this if a wireless network is available.
 - Internet: Wireless
 - LAN: Wired or wireless

Wireless Access Point: Select this if a router already exists on the network.
 - Internet: Wired
 - LAN: Wireless

Local Connection: Select this if you want to create a network without an Internet connection.
 - Internet: None
 - LAN: Wired or wireless

Select the SSID <robot>-5 (if 5Ghz) or <robot>-2 (if 2.4ghz)

(Of course, changing <robot> to the name of the robot)

Set an SSID

Enter an SSID (wireless network name) for the AirStation.

SSID:

Use “quackquack” as the password:

Wireless 5 GHz Security Settings

Select an encryption method. WPA or WPA2 is recommended for protecting your data from unauthorized access.

Encryption Method:

Pre-shared Key Format:

Pre-shared Key:

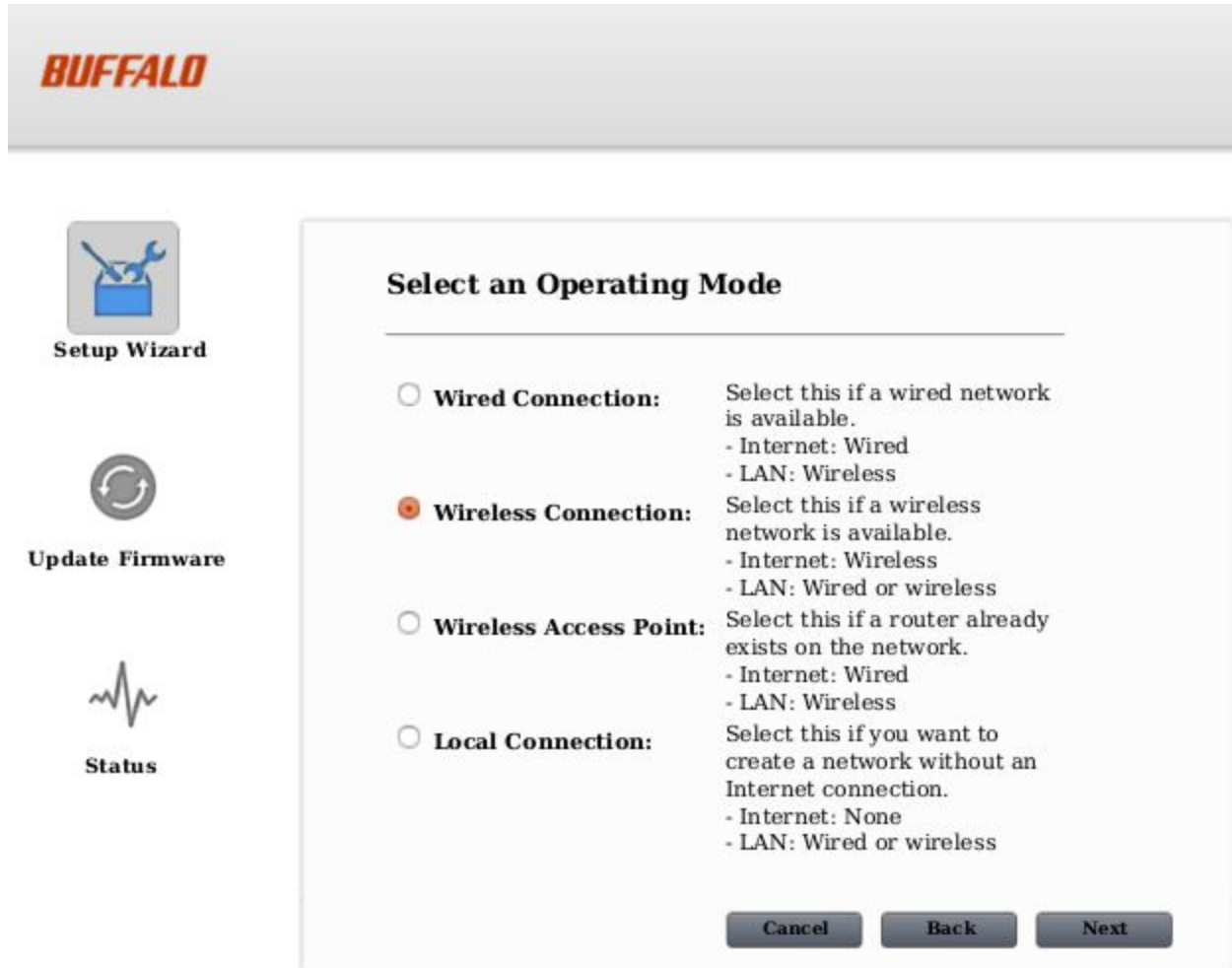
Click **finish** and select **reboot later** in the next screen.

Remember to add “1” to the corresponding column [in this lab document](#).

Step 9: Execute “Internet connection configuration”

NOTE: This is the new step that we added since last time.

Go to the setup wizard once again. This time we will select **Wireless Connection**.



In the next screen click **Search** and select the network that will be your internet connection (duckietown-5).

Click **Next**, select your encryption method (no encryption for many of these networks) and click **Finish**.

The Buffalo will now share this internet connection with both the Pi and your computer. You may have to repeat this step (and only this step) when you move between buildings.

Remember to add "1" to the corresponding column [in this lab document](#).

Step 10: Verifying the network configuration

Wait for the Buffalo to reboot. Then connect to the network <robot>-5.

On your laptop, **but not on the virtual machine**, you should be able to ping the router:

```
$ ping 192.168.13.1
PING 192.168.13.1 (192.168.13.1): 56 data bytes
64 bytes from 192.168.13.1: icmp_seq=0 ttl=64 time=217.857 ms
64 bytes from 192.168.13.1: icmp_seq=1 ttl=64 time=132.172 ms
```

If this works, put a checkmark on the column:

Buffalo			
Name of initial Buffalo network	Buffalo password	Buffalo MAC	can connect to network <robot>-5 and ping 192.168.13.1 from your laptop
Buffalo-A-7582	27833193	cce1d5307582	✓

At this point you should also be able to ping your robot:

```
ping <robot>.local
```

and this should resolve to a 192.168.13.X address.

You should also be able to connect to the internet from both your computer and robot. Try: ping google.com from both your computer and Pi.

Remember to add "1" to the corresponding column in this lab document.

REMOVE THE CANAKIT

REBOOT THE PI

Step 11: Configuring the laptop

There are three sections:

- 1) [10a: configuration for Ubuntu Native](#) - this should be painless and work right away
- 2) [10b: configuration for Virtualbox](#) - this should be painless and work right away
- 3) [10c: configuration for VMWare](#) - **unfortunately there is a bug with the support for multicast; the mDNS (*.local names) does not work, and we will need to add fixed IPs in the /etc/hosts file.**

Step 11a: Configure Ubuntu native

There are no further settings to be done if you are using Ubuntu native.

Connect to the network <robot>-5 or <robot>-2.

You should be able to:

- 1) Ping the router:

```
ping 192.168.13.1
```

- 2) Ping the robot:

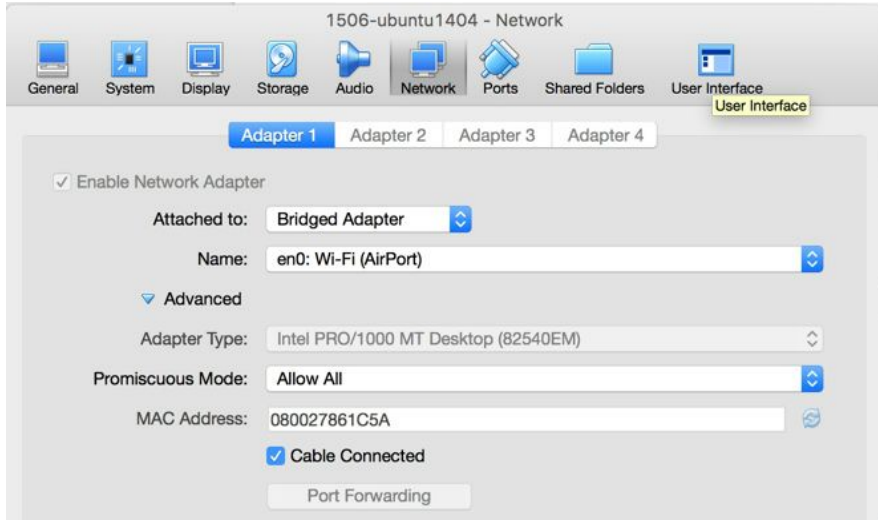
```
ping <robot>.local
```

and this should resolve to a 192.168.13.X address.

Remember to add "1" to the corresponding column [in this lab document](#).

Step 11b: Configure Ubuntu/Virtualbox

For Virtualbox, you should use these settings:



Reboot the virtual machine.

You should be able to:

2) Ping the router:

```
ping 192.168.13.1
```

2) Ping the robot:

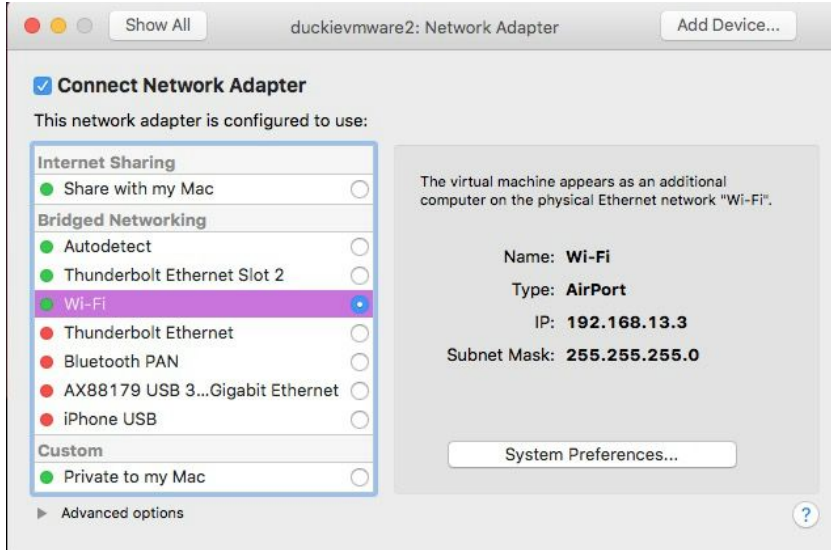
```
ping <robot>.local
```

and this should resolve to a 192.168.13.X address.

Remember to add "1" to the corresponding column [in this lab document](#).

Step 11c: Configure Ubuntu/VMWare

Use "Bridged networking". The IP should look like something like "192.168.13.3":



From Ubuntu, the configuration should look something like this:

```
$ ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:6f:7a:96
          inet addr:192.168.13.3  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe6f:7a96/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11087 (11.0 KB)  TX bytes:20221 (20.2 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:311 errors:0 dropped:0 overruns:0 frame:0
          TX packets:311 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22145 (22.1 KB)  TX bytes:22145 (22.1 KB)
```

You should be able to:

3) Ping the router:

```
ping 192.168.13.1
```

2) Ping the robot using IP:

```
ping 192.168.13.2
```

In practice, we have observed that the IP of the robot given by the Buffalo is always 192.168.13.2, because it is the first one to connect to the Buffalo. If this is not true in the future, change the IP above with the new IP address.

You should not be able to ping the robot by name. This is a **VMWare bug**. Multicast does not work properly.

So this **should fail**:

```
ping <robot>.local
```

The **workaround** is to add the robot name in the /etc/hosts file in VMWare:

```
127.0.0.1    localhost
127.0.1.1    <virtual machine name>

192.168.13.2 <robot>.local
```

At this point, you should be able to

```
ping <robot>.local
```

xRemember to add "1" to the corresponding column in this lab document.

Appendix; checklist

(to finish - don't read)

```
duckiebot $ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr b8:27:eb:8f:bd:8b
          inet addr:192.168.13.2  Bcast:192.168.13.255  Mask:255.255.255.0
          inet6 addr: fe80::ba27:ebff:fe8f:bd8b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3219 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2810 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:316338 (316.3 KB)  TX bytes:1097093 (1.0 MB)
```

```
duckiebot $ ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 00:0f:60:06:0a:4d
          inet addr:10.0.1.12  Bcast:10.0.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20f:60ff:fe06:a4d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:294 errors:0 dropped:0 overruns:0 frame:0
          TX packets:55 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:46745 (46.7 KB)  TX bytes:9463 (9.4 KB)
```

```
duckiebot $ route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0        10.0.1.1       0.0.0.0         UG    0      0      0 wlan0
10.0.1.0        0.0.0.0        255.255.255.0   U     0      0      0 wlan0
192.168.13.0   0.0.0.0        255.255.255.0   U     0      0      0 eth0
```

Note: you might need to do this:

```
sudo route delete default gw 192.168.13.1
sudo route add default gw 10.0.1.1
```