**Basic networking setup**

With a router and the raspberry pi Raspbian, or Ubuntu/Lubuntu

First, what is networking? There are many kinds of network infrastructure, but the by far most common is the one we call the IP protocol, or just internet protocol. Essentially what we are doing here is setting up a tiny internet of PIs, and of course the reason to network the PIs is so they can communicate with one another! We will use this communication later as a basis of a cluster, a bunch of computers working together to solve a problem. The easiest, least expensive and fastest way to network a bunch of computers is using the Ethernet hardware (which has its own low level protocol) to communicate “the internet way” between machines. Most clusters, therefore, are little (or big) local internets!

Rasberry PI’s use DHCP by default the dynamic host resolution protocol. When you plug an ethernet cable into a computer (and the other end of that cable into a switch or router), you want it to more or less automatically connect to the internet, right? What this means is that the router assigns you an arbitrary IP address (four octets separated by the “.” character, say 192.168.0.100 or 10.100.0.100), and associates that local area network address with the hardware address (MAC address) of your ethernet card, which looks something like this: 32:1c:17:81:22:d5. The four octets of an ip addresses are notated in decimal with periods, while the 8 octets of a MAC address are notated in hexadecimal and separated with colons. What is an octet? Just an 8-bit number from 0 to 255, which can be represented in decimal from 0 to 255, and hexadecimal from 0 to FF. None of this math is very important to understand for basic network set up, other than recognizing the difference between an IP address and a MAC address, and sometimes finding the correct numbers.

DHCP does the task for us, keeping the details away from our eyes. Plug in the cable and it works. But for a simple cluster of PIs we prefer a “static ip” to a dynamically allocated one. This is so each computer on the network can share a directory with names and ip addresses that help them find every other computer on the network. In a cluster, the nodes need to know each other and be able to talk to each other in a consistent way. So what we are doing here is setting up static ip addresses on a local area internet. Typically this involves editing some files in /etc.

**Setting up a network**

This tutorial assumes you are setting your network up using a typical consumer running the popluar, open source OpenWrt operating system. See www.openwrt.org. The advantages art that it is that this is normally very easy, very cheap (used wired routers can be had for less than $10 used) and also relatively secure, allowing your network/cluster to live safely behind a firewall (NAT) with easy control over how your nodes access the internet (and how users out there on the internet can connect to your computers.) Or given the possible/pending death of net neutrality, you might be interested in creating your own networks for your home, building, neighborhood, even city.

1) Give your computer a name better than “raspberrypi”. Note that this does not refer to the account name (user “pi”) but the name of your machine as it presents itself to whatever network it is connected to. It is important that all machines connected to your network have a unique name.

$ cat hostname # just displays your computer’s name

(probably “raspberrypi” given the normal Raspbian image…)  
$ cat /etc/hostname

$ sudo (your favorite editor) /etc/hostname # and change from raspberry pi to (whatever you are naming it…) Note: don’t use spaces. In general.

You can also:

$ sudo hostname *somethingelse* # temporary change

$ hostname # should now show your hostname is *somethingelse*

But that is temporary! Then

$ sudo reboot # need to reboot at some point, maybe not now.

…

2) Edit /etc/dhcpcd.conf . This file contains information that your Pi uses to use the network. Scroll down to the section:

# Example static IP configuration:

And make the following *kinds* of changes:

interface eth0

static ip\_address=192.168.1.101/24

#static ip6\_address=fd51:42f8:caae:d92e::ff/64

static routers=192.168.1.1

static domain\_name\_servers=192.168.1.1 8.8.8.8

Notes: you might use other dns servers, see the configuration of your router or other institutional info in order to utilize the dns server provided by your isp or other provider. 8.8.8.8 and 8.8.8.4 are Google’s open dns servers, and they work fine for free, thanks google.

The above configuration is “boiler plate” that you might have to alter a little. (For example, your router’s network address is 192.168.2.1 your PIs should use an 192.168.2.x address in the hosts and dhcpcd.conf files.) The only value we change is to give OPi1 an (arbitrarily chosen) ip address of 192.168.0.100 such that it matches what if found in /etc/hostname. (see the address line.) Again, each Pi on the network needs its own IP address.

Note here that the setup is a little different from the “Raspberry Pi” instructions we offer or that you might find, but it is important to point out that the differences have almost nothing to do with “Raspberry Pi vs Orange Pi”, or better said, “Raspbian vs Lubuntu.” For the ip addresses we are using different reserved ip address space for private networks, because in this example we are running the cluster on something more similar to the private network address you are more likely to find behind the kind of router that you would normally find connected to a home network. (As opposed to a simple switch sans routing in the Raspberry Pi example.) In this example, we are using an old Netgear FVS318 router, and it’s default ip address for the LAN is 192.168.0.1. So, that is the appropriate address to use for both the gateway and network setting in the /etc/network/interfaces file. As a big bonus, the router itself provides us with Network Address Translation (it is built in and managed through the router’s web interface!), which saves us from needing to configure NAT or some other firewall. A major limitation exists only if you plan to connect some hundreds of nodes, but the space of 253 addresss between 192.168.0.1 and 192.168.0.255 (reserved for broadcast) is more than enough for our five planned nodes.

In summary the above interfaces file example hardcodes this OPi to use the ip address 192.168.0.100, and on such a LAN the netmask of 255.255.255.0 and broadcast of 192.168.0.255 are also common values. Most of the time, your values will be similar given home gear. For example, if your router’s LAN address is 192.168.11.11 (default values are set by the manufacturer and are often printed on the outside of your router) then you would use that value for network and gateway, and the same broadcast and netmasks as above, and your choice of ip address on the LAN. Each of your Orange Pi nodes will need a different value for only ipaddress: after 192.168.0.100, we use 192.168.0.101, 192.168.0.102, 192.168.0.103, 192.168.0.104 for the Orange Pi PCs, to correspond to our chosen hostnames for each of the devices opi0, opi1, opi2, opi3 and opi4. The numbering and naming correspondence is for no reason other than mental convenience.

3) Set up your hostname in your hosts table:

$ sudo (your favorite editor) /ect/hosts # and change the localhost entry from raspberrypi to (whatever you have named it…)

The hosts table provides a (static) way for your machine to give names to the other machines on the network, including, itself.

Note (as pointed out before) each of your nodes - each board/each Pi on the network - should have a different hostname. For example, you might plan for the following hostnames for each of your PIs: pi0, pi1, pi2, pi3, pi4…

The IP address indexed by your node name should be the same as your IP address in dchpcd.conf.

One more thing to note: how can you give a name to another pi on the network? Each Pi has a unique hostname and an IP address. (They should anyway.) So /etc/hosts should also have entries for other PIs on your network!

127.0.0.1 localhost RPi0 # local and hostname of this RPi

198.162.0.100 RPi0 # also, this is the local RPi. See /etc/dhcpcd.conf

198.162.0.101 RPi1 # the rest are other nodes on your LAN

198.162.0.102 RPi2

198.162.0.103 RPi3

198.162.0.104 RPi4

You should be able to use the ping command like this and see the other nodes on your network:  
  
$ ping RPi2

**Rasperry Pi specific stuff**

Raspian does not have the ssh daemon turned on by default. You will have to use the raspi-config utility (with sudo) to turn it on. Then you should be able to ssh to the other PIs on the network, assuming you have their password of course.