

## 2: Setting up your first node

Section	Length
<a href="#">1: Topic introduction</a>	15 min
<a href="#">2: Hands-on and material exploration</a>	40 min
<a href="#">3: Discussion and reflection activity</a>	20 min
<a href="#">4: Wrap-up and week ahead</a>	5 min
Workshop length	1:30 min

### 1: Topic introduction

- Recap that in the last session, we ran a peer-to-peer application over a mesh network isolated from the internet, and today we will manually set up such a mesh network on the same Raspberry Pi devices running GNU/Linux
- Summarize homework takeaways and answer any questions:
  - IP address and packet
  - DHCP, NAT, IPv6 self-addressing
  - Packet forwarding and network routing
- Act as DHCP server and assign IP addresses to a couple students, route a paper message, using the terminologies, to demonstrate these concepts

### 2: Hands-on and material exploration

Make visible the underlying infrastructure by sending text messages over ethernet cables.

#### Objectives

Set up a very basic network in class as manually as possible with the following goals:

- Allow students to relate familiar day-to-day user interfaces (i.e. a desktop browser or a mobile app) to the physical infrastructure that carry our communications (i.e. an ethernet cable or a wireless radio)
- Allow opportunity for students to tinker with command line tools, and help each other out to encourage an atmosphere of peer-to-peer learning, since we expect varying degrees of experience and comfort with typing into a shell from our audience (small set of simple commands with recoverable states)
- Concretize abstract concepts discussed earlier such as *IP addresses* and *centralize points of failure in hierarchical networks*
- As we go through the steps in the activity, make a connection between what we are building to concepts that we have discussed before

#### Materials

- An even number of Raspberry Pi 3s and power supplies
- SD card with [mesh-workshop \(https://github.com/benhylau/mesh-workshop\)](https://github.com/benhylau/mesh-workshop) pre-configured to mesh wirelessly with cjdns
- One ethernet cable for each pair of Raspberry Pi 3s

- Two USB WiFi radios that is [ad-hoc or 802.11s-capable \(https://github.com/phillymesh/802.11s-adapters\)](https://github.com/phillymesh/802.11s-adapters)
- A 4-port network switch (to demonstrate network centralization)
- A physical lock with a key (to demonstrate end-to-end encryption)

## Format

- Groups (or Pairs) to start, then Class

## Activity

### Make wired ethernet link and assign route with ip

- Each group starts with a pair of Raspberry Pis
- Each group has two laptops, each connecting to one of their Raspberry Pi's Host AP:
  - Connect to SSID **<hostname>** with password **password**
  - Initiate a SSH session to **root@<hostname>.local** with password **root**
- In the SSH session, look at the **eth0** ethernet interface using **networkctl status eth0**
- Connect the two nodes with an ethernet cable and run **networkctl status eth0** again
- Explain that we now have a road (i.e. ethernet cable) but no street addresses (i.e. IP addresses) or road signs for directions (i.e. routes)
- Assign an IP address to **eth0** and add a route on **<host1>**:

```
root@host1:~# ip addr add 192.168.0.1 dev eth0
root@host1:~# ip route add 192.168.0.0/24 via 192.168.0.1
```

and on **<host2>**:

```
root@host2:~# ip addr add 192.168.0.2 dev eth0
root@host2:~# ip route add 192.168.0.0/24 via 192.168.0.2
```

- Verify the results with **networkctl status eth0** and **ip route**

### Communicate over the wire with nc

- Use **nc** to send plaintext messages to one another, listen on port 80 on **<host1>**:

```
root@host1:~# nc -l -p 80
```

and connect to it from **<host2>**:

```
root@host2:~# nc <host1>.local 80
```

then start typing text messages

- Run a minimal webserver that can respond to HTTP messages. Look at the script **start-webserver.sh** and run it:

```
root@host1:~# cat ~/scripts/start-webserver.sh
root@host1:~# sh ~/scripts/start-webserver.sh
```

and send a HTTP request to it from **<host2>**:

```
root@host2:~# curl <host1>.local
```

then observe the response from the webserver. When you are ready to move on, hit **Ctrl + C** to stop the server.

- This is similar to manually performing the task of a browser or application

### Form a 3-node centralized network

- End the group stage and take three volunteers to form a network by plugging three Raspberry Pis into the network switch each with an ethernet cable
- Point out two people in different groups may have selected the same IPv4 address for their **eth0** interface, like how two previously disjoint communities assigned the same street name and house number, which only becomes a problem when we bridge these two communities
- Talk about NAT and self-assignment of IPv6
- Resolve IP conflicts and show how we can talk to both other nodes from one node
- Explain that if there is an outage at the switch none of the nodes can communicate anymore, and that is analogous to our Internet Service Providers in traditional network topologies

### Form a 3-node mesh network

- As a class, construct the following network topology to form a mesh network and send a message from one edge node to the other, utilizing the middle one as relay:

```
+-----+ +-----+
+-----+
| <host1> with      | wireless | <host2> with      |
ethernet cable | <host3> with      |
| mesh point adapter +-----+ mesh point adapter + ethernet cable
+-----+ ethernet cable |
| running cjdns      |      | running cjdns      |
| running cjdns      |      | running cjdns      |
+-----+ +-----+
```

- From **<host1>**, ping **<host2>** and then **<host3>**
  - Explain mesh routing over relay **<host2>** across different network interfaces
  - Explain IPv6 self-addressing to prevent IP collision problem we discussed
  - Explain how end-to-end encryption makes the middle node unaware of what is in the data stream
- Using the lock and key, make analogy to mailing a locked package where the recipient has the key to open the lock)

## 3: Discussion and reflection activity

Spend a little time to casually discuss how our internet's Domain Name System (DNS) works, and who controls which Top-Level Domains (TLDs). Draw from [I Think Therefore ICANN](http://networks.land/activities/i-think-therefore-icann/) (<http://networks.land/activities/i-think-therefore-icann/>) by [Networks Land](http://networks.land/) (<http://networks.land/>).

Start with [The "original" TLDs](http://networks.land/reference/top-level-domains/) (<http://networks.land/reference/top-level-domains/>) and emphasize which organization controls each domain.

There will not be enough time to play [I Think Therefore ICANN \(http://networks.land/activities/i-think-therefore-icann/\)](http://networks.land/activities/i-think-therefore-icann/), but we can print some of the "Neat Real-World Examples" on cards and share them with the class, for example:

**.1y:** Libya's two governments, as well as other splinter groups, are competing over the ability to communicate officially via the internet, and in places it comes down simply to who holds the login details: "the government domains are distributed between the two parties and it depends on who has the password which allows them to change the details of the domain."

Hand out the scenario cards each to a different student, when we get to that particular TLD, they can volunteer to read the incident on the card to the class.

## 4: Wrap-up and week ahead

- Recap that we have:
  - Discussed networking acronyms and concepts such as IP addresses, DHCP, NAT, and routing
  - Started off building a simple wired ethernet link across two Raspberry Pi nodes
  - Sent plain text and HTTP messages across that link, emulating how applications communicate under the hood
  - As a class, constructed both a centralized network and a mesh network by hand
- Next session we will discuss wireless links more extensively, both theoretically and the practical aspects when configuring a wireless mesh network

## Homework

- [Frequency, Amplitude, and Phase \(https://www.youtube.com/watch?v=5g-Din357iY\)](https://www.youtube.com/watch?v=5g-Din357iY) from the [Metamesh YouTube channel \(https://www.youtube.com/channel/UCGEentxbGKMU9J9GIZ1LQUQ\)](https://www.youtube.com/channel/UCGEentxbGKMU9J9GIZ1LQUQ)
  - Explains what "is" WiFi in the physical sense
  - Relates radio interference to WiFi channels
  - Shows configuration of WiFi channel that is relevant to real-world network deployments
  - Helps future discussions around urban vs. rural deployments and omnidirectional vs. directional links
- Retrieve video content from Hyperboria server as stretch goal