

test_speed program1_3 LCWA

version: 1.0

Andi Klein
pabloemma@gmail.com

January 25, 2022

1 Overview

test_speed is a python program written by Andi Klein. It's main purpose is to collect data from different raspberry pi's distributed across the LCWA to check the bandwidth over 24 hours. This is strictly a program for **La Canada Wireless Association**, which is a notforprofit volunteer organization , which provides internet access to mostly rural communities around Santa Fe.If you want to use this code for other places, feel free to modify it to suit your needs.

The heart of the code are calls to the **Ookla** speedtest servers and/or the **iperf3** library.

The test gets executed every 10 minutes and the data are stored locally. Every hour the program connects to dropbox, and uploads the datafile and the plot file. At midnight every day, the program flushes all the data to dropbox and exits. The computer then does a git pull to a central repository, performs a pull and the restarts. This ensures that any program updates are caught by the clients. Until 2022, the program was using Ookla's speedtest and reporting those numbers. As is documented below, the user can choose the Ookla server, according to their preferences. Over time it has become clear that we would like to have a speedtest which is strictly within the LCWA network, and not a combination of this network and the internet. Several times we noticed strange test result, which were driven by an overloaded Ookla server and not a degradation of the LCWA network. This led to the development of a new addition to the code using iperf3. This iperf3 connection goes to a LCWA iperf3 server, therefore providing a more honest picture

of what is going on in our network. This program runs on **Mac** and **Linux** but **NOT** on Windows. There are currently no plans to ever port it to Windows.

2 Installation and upgrade

Before you can do anything, you have to clone the repo. Create a directory /git/speedtest with `mkdir -p git/speedtest`.

- `cd git/speedtest`
- `git clone https://github.com/pabloemma/LCWA .`
- if you need a different branch: `git fetch; git branch -v -a; git checkout branchname`

The program has many components, which rely on different python and system packages. I hope I have them all in the install script. If you run in difficulties, send me the output of the installation.

Sometimes I will make changes which require an upgrade, and then the script `upgrade_speedtest` will be run.

2.0.1 Installing on a Linux raspberry pi system

In this case you should be able to use the provided install script : `install—_speedtest`. There is also a mac installation script, but this is currently outdated.

```
cd ../src
./install_speedtest 2>&1 | tee install.log
```

If the installation crashes, please send me the `install.log` file.

2.0.2 general instructions for running the system on your own

In the following a discussion on what you need to install the project:

1. Install the speedtest CLI from <https://www.speedtest.net/apps/cli>
 2. Once installed you need to run it once to accept the license
 3. pip install dropbox
 4. Mac: brew install git, Linux it is usually supplied, otherwise sudo apt get install git , or yum install git
 5. mkdir /speedfiles (below your home directory)
 6. mkdir git
 7. cd git
 8. git clone <https://github.com/pabloemma/speedtest.git> speedtest (this you only do once, after this your command will be git pull , which will update you to the latest version)
 9. You will need a dropbox account and setup an app. If you have a dropbox account, log in and then go to <https://www.dropbox.com/developers/documentation>, click on “App console” in the upper right corner and create a new app. There click on permission type and select which one you want. Once you generated the app, click on the button Generate under “Generated Access Token”
 10. Copy into clipboard the Access token, which is very long
 11. open the file LCWA_d.txt in your speedtest directory, remove the line and paste your access token.
 12. On dropbox also create a folder called “LCWA”, this is where the files will go.
1. install iperf3 (**not iperf or iperf2**)

3 Running and controlling the program

The program has currently two different ways to run; either speedtest, which is a wrapper around the Ookla speedtest or iperf, which will run an iperf test to a server on the LCWA network. The issue with the Ookla server is, that the result is depending on quite a few parameters beyond LCWA's control. The current server is with cybermesa, and we have noticed that the server is sometimes slow. Also, since we go outside onto the internet, the results are influenced by whatever traffic is going on. The iperf test really checks the available bandwidth within the LCWA network, and is therefore a better measure of what we provide.

The program checks every 10 minutes the bandwidth, and records this in a file in the directory `/home/pi/speedfiles`. Every hour around `xx:30` the csv file gets plotted into a pdf file and both are shipped to the corresponding dropbox. At midnight a second program collates all the different speedboxes, and creates a full 24 plot file. Also at midnight the program terminates, performs a git pull for updates, and then restarts again. In order not to blast the servers at the same time, each speedbox has a delay in the startup corresponding to the 2 digits in the name.

$$delay = XX * 25 \quad (1)$$

where the `XX` stands for the two digit speedboxnumber (`LC04`) would mean 04 as the multiplier and leads to a 100 sec delay.

3.1 Required directory settings

In order for me to support the program, I require the following directory structure:

`/home/pi/git/speedtest` is where all the components for the speedtest are located. The data files produced will be in `/home/pi/speedfiles`. If you choose a different configuration, you have to change the config file and you are on your own.

3.2 config file

Here is an example of a config file:

```
{  
  
    "Darwin" : {  
        "timeout" : "/usr/local/bin/timeout",
```

```

        "speedtest" : "/usr/local/bin/speedtest",
        "srcdir" : "/Users/klein/visual studio/LCWA/src/",
        "datadir" : "/Users/klein/speedfiles/",
        "conf_dir" : "/Users/klein/visual studio/LCWA/config/"
    },
    "Linux" : {
        "timeout" : "/usr/bin/timeout",
        "speedtest" : "/usr/bin/speedtest",
        "srcdir" : "/home/pi/git/speedtest/src/",
        "datadir" : "/home/git/speedfiles/",
        "conf_dir" : "/home/pi/git/speedtest/src/config/"
    },
    "Control" : {
        "runmode" : "Iperf",
        "debug" : false,
        "cryptofile" : "LCWA_d.txt"
    },
    "Iperf" : {
        "serverport" : "5201",
        "serverip" : "63.229.162.245",
        "duration" : 10,
        "numstreams" : 2,
        "blksize" : 1024,
        "latency_ip" : "65.19.14.51",
        "time_window" : 10,

        "reverse" : false
    },
    "Speedtest" : {
        "serverip" : "63.229.162.245",
        "serverid" : 18002,
        "time_window" : 10,
        "latency_ip" : "65.19.14.51"
    }
}

```

As you can see , this is a json file (which personally I think is trash; who in their right mind develops a system, which does not allow comments) and we can look at it in some more details:

The first two block deal with the runtime environment. Since Apple and Linux have their stuff not in the same location (thank you apple) , we have to specify the locations of some critical files and directories. The next block **Control** deal;s with issues which are common to both runmodes and operating systems. **Iperf** selects the iperf test, while **Speedtest** controls space aliens (just kidding and making sure you haven't fallen asleep yet).

The last two blocks contain control parameters used for the two different runmodes.

If you choose Both for the running, you should also provide the variable random in the control block and set it to either true or false. If fasle, the program will switch back and forth between iperf and speedtest on a regular basis, indeed every time it should run . If you set random to true, it will decide randomly which way to go.

3.3 Running

Everything is controlled through two crontab entries, one on the user level and one the sudo level. The root level one starts the system after the midnight exit, and the user one restarts the program if for some reaso it exits. Here are the two crontab settings:

The first one you get with sudo crontab -e

```
Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
```

```
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow    command
@reboot sleep 60 ; /home/pi/git/speedtest/scripts/run_speedtest >>/ho
```

The second one is just crontab -e and looks like this:

```
.....
.....
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow    command
*/10 * * * * /home/pi/git/speedtest/scripts/restart.sh

# run every hour at 55 minues
55 * * * * /home/pi/git/speedtest/scripts/git_pull.sh

5923 * * * /home/pi/git/speedtest/scripts/git_pull.sh
```

where I only show the last few lines