test_speed program1_3 LCWA
version: 9.02.03
doc version 1.2

Andi Klein
pabloemma@gmail.com

March 26, 2024

Contents

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 not forprofit 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. Since this is used on a cluster, with nodes which are not easily accessible, after the git pull, you can perform a limited task of maintenance through the **maintain.sh** script. 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 resulst, 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.

As of late 2023, there is no iperf server locally accessible anymore. This is a real shortcoming, since a comparion of local iperfdata with ookla speedserver data would allow us to isolate our real network performance from the internet. Basically subtracting the iperf results from the speedtest data, would deconvolute the two.

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. You can also get to another existing branch by typing git checkout **branch**

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.

2.1 Installing on a Linux raspberry pi system

outdated needs to be rewritten 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.

3 Running 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 bandwith, and records this is 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.

$$delay = XX * 25 \tag{1}$$

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 Controlling and the config file

Here is an example of a config file:

```
{

    "Darwin" : {
        "timeout" : "/usr/local/bin/timeout",
        "speedtest" : "/usr/local/bin/speedtest",
        "srcdir" : "/Users/klein/git/LCWA/src/",
        "datadir" : "/Users/klein/speedfiles/",
```

```
    "logdir" : "/Users/klein/git/speedtest/log/",
    "conf_dir" : "/Users/klein/git/LCWA/config/",
    "doc_dir"     : "/Users/klein/git/LCWA/doc/"

},
"Linux" : {
    "timeout" : "/usr/bin/timeout",
    "speedtest" : "/usr/bin/speedtest",
    "srcdir" : "/home/pi/git/speedtest/src/",
    "datadir" : "/home/pi/speedfiles/",
    "logdir" : "/home/pi/git/speedtest/log/",
    "conf_dir" : "/home/pi/git/speedtest/config/",
    "doc_dir"     : "/home/pi/git/speedtest/doc/"
},
"Control" : {
    "runmode" : "Speedtest",
    "debug"   : false,
    "cryptofile" : "LCWA_a.txt",
    "click"      : "1",
    "random"     : false,
    "keep_files_time" : 7,
    "server_file" : "ookla_us.csv",
    "server_list" : { "nmsurf":9686,"plateau":16869,"comcast":1773,"sa

},
"Iperf" : {
    "iperf_serverport"  : "5201",
    "iperf_serverip" : "63.229.162.245",
    "iperf_duration"  : 10,
    "iperf_numstreams" : 2,
    "iperf_blksize"   : 1024,
    "iperf_latency_ip" : "65.19.14.51",
    "iperf_time_window" : 10,

    "iperf_reverse"   : false
},
"Speedtest" : {
    "serverip" : "63.229.162.245",
    "serverid" : 9686,
    "time_window" : 10,
```

```
    "latency_ip" : "65.19.14.51"
 },
"ClusterControl" : {
    "LC01" :  {
        "runmode" : "Speedtest",

        "nondefault" : {

                "server_ip" :"63.229.162.245" ,
                "serverid" : 9686,
                "time_window" : 10,
                "latency_ip" : "65.19.14.51" ,
                "iperf_serverport"  : "5201",
                "iperf_serverip" : "63.229.162.245",
                "iperf_duration"  : 30,
                "iperf_numstreams" : 2,
                "iperf_blksize"   : 1024,
                "iperf_latency_ip" : "65.19.14.51",
                "iperf_time_window" : 10,

                "iperf_reverse"   : false,
                "random" : false
                 },


        "logging" : {
                "loglevel" : "info",
                "output"   : "info.log",
                "log_conf_file" : "logger.json"
           }
        },
```

As you can see , this is a json file (which personally I don't like; 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** deals 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 false, 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.

I added a clustercontrol block, where you can fine tune the parameters for every speedbox. If you don't change anything in the clustercontrol, it takes the default values.

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 >>/home/pi
```

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

59 23 * * * /home/pi/git/speedtest/scripts/git_pull.sh
```

where I only show the last few lines

4 Results

Below is the typical output of one of the speedboxes for a 24 hr period. During this time, Lumen had huge problems with delivering our bandwidth and this is reflected in the black squares, which measure the download speed to a speedtest server. The red triangles are the uploadspeed to the same server. In contrast the blue and green are our iperf results, which test the bandwidth within our network.

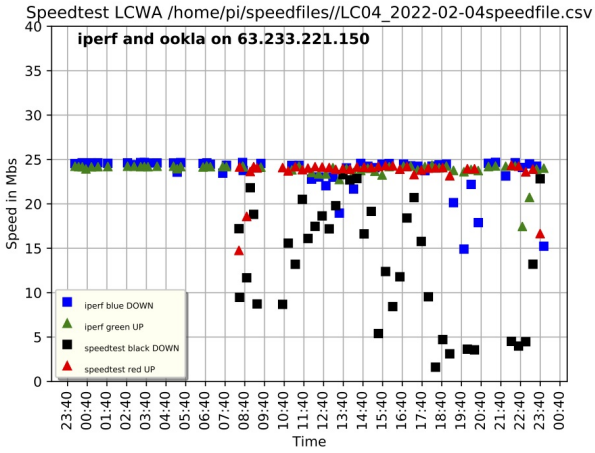

Figure 1: LC04 on Feb 04 2022

There are four files produced with extensions pdf, csv and txt and log. The csv file contains the raw data , the pdf is a plot of the data and the txt file has a list of parameters the program run under, plus at midnight a statistics output. The last file (log), records all logging messages from the system. There are three levels; info, warning and error. In the following the midnight output from the plotted run. The important info are the run conditions at the beginning, which in this case is **Both** , and then the different parameters for iperf and speedtest.

```
IP    63.233.221.150
Date    2022-02-04 23:50:47.244401
Dropbox    /LCWA/LC04_/
MacAddress    b8:27:eb:41:62:97
File    LC04_2022-02-04speedfile.csv
version    8.02.01
runmode    Both
iperf server    63.229.162.245
iperf port    5201
iperf numstreams    2
iperf blocksize    1024
iperf duration    30
iperf reverse    False
time window    600
ookla server id    18002
```

```
latency ip    65.19.14.51
random    False


 **********************************total statistics**********************
Iperf
Min download                  = 14.903761103996434
Max download                  = 24.66901502375453
Mean download                  = 23.558043052947266
Std download                  = 2.128372106266123
Min upload                 = 17.440534244451296
Max upload                 = 24.34935391081165
Mean upload                 = 23.83621240450195
Std upload                 = 1.015410482611458


Ookla Speedtest
Min download                  = 1.598744
Max download                  = 23.282152
Mean download                  = 13.355808000000001
Std download                  = 6.769985371569937
Min upload                 = 14.74664
Max upload                 = 24.351616
Mean upload                 = 23.40635557894737
Std upload                 = 2.067538460635904


 **********************************end statistics**********************
```