

Master Course Computer Networks

Project: MeasrDroid

Last Update: November 11, 2013

General Information

This project is about Internet measurements: you will learn about the Internet's dynamic topology, asymmetric routing and environmental influences in mobile networks. Your tasks in this project are 1) to use an Android-based measurement application developed at our chair, 2) to understand and complement its measurements and 3) to evaluate the collected data.

MeasrDroid is a free measurement client with focus on network research. It performs a variety of active network measurements, and provides additional data about the status of the device and its environmental conditions. For this project, you have to download and install a modified version at

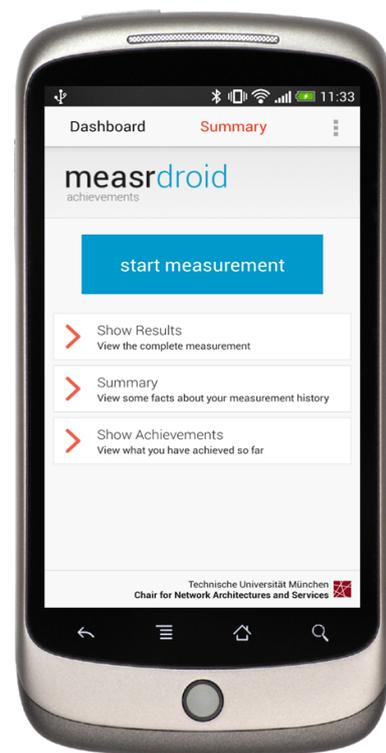
<http://mccn.droid.net.in.tum.de/>

Follow the configuration wizard and accept the privacy agreement. You may configure the measurements to your needs, but we recommend to leave all measurements activated, and to donate at least 50 Megabyte of monthly traffic volume. The last configuration step asks you to enter your team number in the format `tXX`. After successful completion, the **MeasrDroid** app will start to periodically perform measurements, which will be automatically uploaded to your virtual machine. Further information on how to access your virtual machine will be made available on the course website¹.

After the team selection process is over, you will be provided with a shared folder in the course's subversion repository. Your project plan and your final report have to be committed into this folder before the deadline. The remaining deadlines will be enforced as described below.

You have to use the `python` language for all programs to be developed in this project. Adhere to the python style guide PEP² and don't forget to comment – your code quality is part of the grade. You can use the `pylint` tool to get an automatic rating of the quality of your code and suggestions for improvements.

Note: This project requires basic experience in working with Linux and using a shell. We recommend that at least one team member has some previous experience.



¹<http://www.net.in.tum.de/en/teaching/ws1314/vorlesungen/master-course-computer-networks/>

²<http://www.python.org/dev/peps/pep-0008/>

1 Project Plan – November 12, 2013

[1 point]

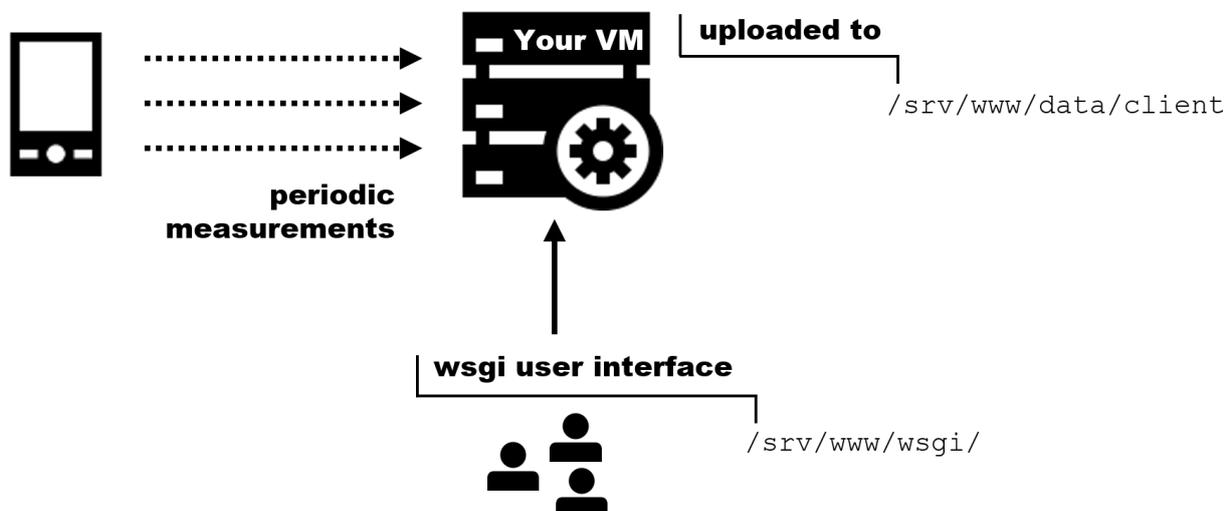
The goal of creating a project plan is that your team has a common understanding how you want to handle the project. Read all the tasks of the project and discuss them conceptually. How do you plan to split the work among yourselves? How much time do you plan to dedicate to the project? Are there specific difficulties you already anticipate?

We expect you, as a team, to exchange thoughts and think about the project. The output of this task should be about one page of text, possibly including a Gantt chart to outline your schedule. The point will be given to you if this document shows that you have a clear conception of how you will proceed and a mutual understanding regarding the above questions.

2 Monitor Measurements – December 3, 2013

[1 point]

Make yourself familiar with the measurement data uploaded by your MeasrDroid client. You will create some basic plots and publish them as a web-based live view. The architecture for this task is as follows:



a) Investigate uploaded data sets

Locate the MeasrDroid data sets and investigate the data manually. It is encoded in JSON format and is thus human-readable. Identify relevant data points for delay measurements and make yourself familiar with the JSON interface in `python` to load the data sets and extract the measurement data.

b) Plot delay measurements

Use the `python` library `matplotlib` to create a *time-series-plot* and a *box-whisker-plot* for the delay measurements. You should include meaningful axis labels.

c) Provide a live view

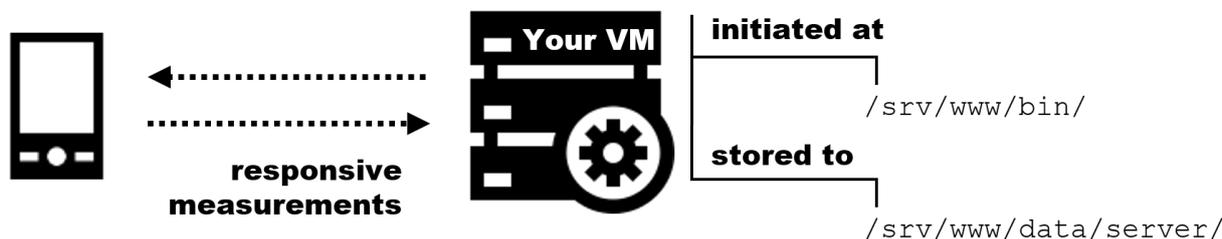
Adopt your plotting code to be served as a `WSGI-script`. A *Hello World* example can be found in the `wsgi`-folder on your virtual machine. You can access it with your browser by navigating to <http://tXX.droid.net.in.tum.de/wsgi/hello>. However, you might want to extend the webserver's configuration file located at `/etc/apache2/sites-available/droid.conf`.

Once the deadline is reached, we will navigate to your website and check if plots are online and updated dynamically.

3 Initiate Measurements – December 17, 2013

[2 points]

Develop a service to respond to incoming measurements. After successful completion this task, every incoming measurement triggers a similar measurement towards your MeasrDroid client. The architecture for this task is as follows:



a) Investigate incoming measurements

Use the `tcpdump` tool to investigate incoming measurements. You may press the measurement button in MeasrDroid to initiate a manual measurement. Identify incoming packets and attribute them to delay and path measurements. Hint: MeasrDroid uses *ICMP echo requests* and standard Linux `traceroute`.

b) Respond to measurements

Develop a daemon service to listen for measurements. Every incoming measurement should trigger exactly one corresponding outgoing measurement. Store the data in JSON format similar to the uploaded MeasrDroid data, and add plots for outgoing delay measurements to your live view. Once the deadline is reached, we will check your website for new plots, and issue measurements towards your virtual machine to check for counter-measurements.

4 Evaluation – January 21, 2014

[4 points]

Further analyze your collected data to identify environmental influences and to learn about asymmetries in network paths. You may create additional analysis scripts to support this task.

a) Analyze environmental influences

Re-check the uploaded MeasrDroid data to identify environmental conditions that could have influence on the delay measurements. Promising candidates for such data points might be signal strength (WIFI/mobile), battery level and temperature, CPU and RAM utilization, and data stemming from environmental sensors for magnetic field strength, pressure, etc. Plot this data and add it to your live view for comparison with your delay plots. In addition, compute *correlation coefficients* for incoming and outgoing delay measurements, as well as for incoming delay measurements and the environmental data. Describe and interpret your findings.

b) Analyze path asymmetry

Develop metrics to analyze path asymmetry in traceroute measurements. You should reasonably consider paths of different lengths as well as on different abstraction layers (IP addresses vs. autonomous systems). A good starting point could be the concept of *Levenshtein distance*. Describe your approach, interpret your findings and add the analysis to your live view.

Once the deadline is reached, we will check your website for new plots, corresponding correlation coefficients and metrics for path asymmetry.

5 Final Assessment – February 4, 2014

[2 points]

Create a report about the project.

a) The first part of the report should include your results along with meaningful plots, and interpretations of your findings. You should specifically answer the questions raised throughout the tasks above. The scope of this part is about two pages of text (excluding plots).

b) The second part is about reflecting on your work and about giving feedback. You should address the following questions:

- How much time did you invest into the project?
- Was it too easy/too difficult/just right?
- What did you like/dislike about the project?
- What would you do differently now?
- What would you tell people who still had to do this assignment?