Action plan

Rolf Jagerman, Laurens Versluis and Martijn de Vos

May 1, 2014

1 Introduction

This report is part of the bachelor end project on anonymous video streaming on tablets. In this project we attempt to get Tribler, a peer-to-peer file sharing application working anonymously on android using an already implemented Tor-like protocol.

Tribler has been developed at Delft University of Technology, by the Parallel and Distributed Systems group. The research performed by this group is focused on designing, implementing and analyzing parallel and distributed systems. As most of the research in this field is empirically conducted, prototypes and real working software are extremely important. Tribler is a peer-to-peer file sharing tool that is completely decentralized. It is used as a vehicle for research on parallel and distributed systems. Tribler has been in development for over 9 years and has a very mature and well-established code-base. An anonymous networking protocol based on Tor and implemented in Python is part of a new beta version of Tribler.

In this report we outline the details of the assignment. We will describe the approach we will use and how we will structure this project. Finally a quality assurance is given.

2 Assignment

In this section we will describe our assignment, client, contacts, the problem we will address and sketch out what product we will eventually deliver. This section will also contain some of the critical requirements that we will have to meet along with the risk involved.

2.1 Assignment

Our assignment is to implement a new feature of Tribler into a mobile android application. This new feature allows the creation and usage of so called Tor-tunnels. These tunnels allow anonymous download within a Peer-to-Peer network between devices, in our case mobile devices. As these tunnels run on

Python code, we will have to be able to run Python on an Android application, along with all the libraries it depends on.

2.2 The client

Our client is Dr. Ir. Johan Pouwelse, the head of the Tribler group and Assistant Professor at the Parallel and Distributed Systems Group of the Faculty of EEMCS, Delft University of Technology. Pouwelse has measured and researched Peer-to-Peer networks for years and has been working for nine years on Tribler.

2.3 Contacts

The Client / Project Coach:

Dr. Ir. Johan Pouwelse
J.A.Pouwelse@tudelft.nl
+31 (0)15 27 82539
Room: HB 07.290
Mekelweg 4
2628 CD Delft

Project Advisor:

Egbert
?
?
Mekelweg 4
2628 CD Delft

Bachelor Project Coordinator:

Dr. Martha A. Larson
M.A.Larson@tudelft.nl
+31 (0)15 27 87357
Room: HB 11.040
Mekelweg 4
2628 CD Delft

2.4 The final product

At the end of the Bachelor Project, we will deliver an Android application that allows users to find and download content from other relays (mobile devices). This application makes use of the code from the Tribler project, which is written

in Python. Therefore, we will make sure that the Android application will be able to run Python code.
If all is well, the downloads that run through the Tor tunnels will be anonymous, just like the current Tor protocol.

2.5 Requirements and risks

The final product will offer the features that are described in the previous section and should be considered a prototype. The prototype will offer anonymous downloading using the Tor tunnels mentioned earlier. The development of the prototype will be targeted to the Android platform.

The following requirements are set during the project:

- Weekly scrum evaluations. At the end of each scrum iteration, we evaluate what we have done and set the target for next week. This keeps the deadlines SMART and manageable.
- Weekly meeting with the client. Every two weeks we will implement a feature, but we will show and discuss progress each week with the supervisor to ensure we are doing the right thing and are still on schedule.
- The members of the team will complete a prototype at the end of this project and will also demonstrate this during a 30 minute presentation given in the last week of Q4.

The risks involved with this project include:

- Run Python code on an Android application. As the Tor-tunnel functionality is written in Python code, we need to be able to run Python code on an Android device as well. A library exist where you can write Python for Android, but we will still face a challenge when we will try to combine other libraries.
- As we are dependent on third party code, we might lose time to understand or read certain parts of code or documentation as well as link pieces of code that belong to different parties together.

3 Approach

In this section, we will describe the approach we are taking for this project. First, we will discuss the methodology we are using (Scrum). After that, we will discuss the MoSCoW technique we are using to classify requirements. Afterwards, we will give an overview of the tools we will be using during this project. Finally, we will give our planning and milestones.

3.1 Scrum methodology

For the project, we will be using the Scrum methodology. We have used Scrum in various projects already during our study and it has proven to be a very efficient way of working. In this section, we will discuss how we plan to use Scrum and how our scrum iterations will look like. first we will look at the different roles involved in the Scrum process. After that, we will describe how the Scrum process is organized.

3.1.1 Scrum roles

There are three primary roles involved in Scrum:

- Product owner: the product owner represents the stakeholders and is the voice of the customer. He is responsible for the success of the product. Johan Pouwelse is the product owner.

- Development team: the development team consists of Rolf, Laurens and Martijn. We are responsible for delivering the final product to the product owner.

- Scrum master: he guides the team by assuring the right choices are being made. He is responsible for arranging the meetings. Rolf is our scrum leader.

3.1.2 The Scrum process

There are several steps involved in the Scrum process. First, we will create a product backlog. This is a list with the functional demands the product owner has and it contains the items we still have to do. In total, we have 5 sprints. At the end of each sprint, we will deliver a part of final product. The duration of each sprint is two weeks. At the beginning of each sprint, we will create a sprint backlog. This backlog describes the functional demands, divided in each subtask for this sprint.

Each morning, we will start the day with the daily scrum. This is a short meeting where every member of the development team answers the following question:

- What have you done yesterday?

- What are you going to do today?

- Are there any problems you ran into?

At the beginning of each sprint, we start with a meeting. This meeting is attended by all members of the team and the supervisor. The purpose of this meeting is to evaluate the last sprint and decide on the tasks that have to be done during the next sprint.

3.2 MoSCoW

MoSCoW is a technique that can be used to place importance on the delivery of each requirement. In MoSCoW, we have the following categories:

- Must have: the requirement must be part of the final product to be considered a success.
- Should have: the requirement has a high priority that should be in the final product if possible.
- Could have: the requirement is desirable but not necessary.
- Would have: the requirement is not implemented in a given release but is considered as a requirement in the feature.

3.3 Tools

For the project, we will use various tools, both hardware and software. First of all, we will be using our own laptops for the development of the software. We will develop our software on the Ubuntu platform. We are also using Android phones that we can rent from the Tribler team.

If we look at the software, we will make use of the Android SDK and NDK. The Android SDK will allow us to build .apk files. The NDK allows to implement parts of an Android application in C or C++. To be able to run Python code on an Android device, we will use the Python for Android library.

3.4 Planning

Our planning can be found on Github. As for now, we have four milestones:

- 02-05-14: the literature research should be done and a report about the read literature should have been written.
- 09-05-14: we should be able to compile the TGS for Android project and run it on an Android device.
- 30-05-14: be able to send a packet between two Android devices over anonymous tunnels.
- 13-06-14: designed and created a GUI for testing purposes.
- 27-06-14: if everything goes well, this date we should have worked together with the other bachelor end project group to enable video streaming over anonymous tunnels.
- 04-07-14: the end presentation of our project.

4 Project structure

In this chapter the administrative aspects of the project are described.

4.1 Members

The members of the project are Rolf Jagerman, Laurens Versluis and Martijn de Vos. All members will work 40 hours per week to ensure the mandatory 15 EC per student are utilized. The division of labor is evenly distributed across all activities (analysis, documentation, implementation, etc.).
Contact Information
Rolf Jagerman - R.M.Jagerman@student.tudelft.nl
Laurens Versluis - L.F.D.Versluis@student.tudelft.nl
Martijn de Vos - M.A.Devos@student.tudelft.nl

4.2 Reporting

Weekly meetings with the client will be held in person. The notes of these meetings will be recorded and stored online. All documentation of the project will be written in LaTeX and will be provided as a PDF. The project material, including source code and documentation, will be available throughout the project on a GitHub repository: www.github.com/rjagerman/AT3.

5 Quality assurance

To assure a good quality of the delivered product, several agreements are made about which methods should be used. In particular we look at documentation, testing, code review and version control.

5.1 Documentation

All documentation regarding the software will be written as Python Docstrings. This will be parsed and exported into online documentation. Additional tutorials or information will be supplemented in this online documentation.

5.2 Testing

All written software will be tested using python unit tests. Additionally, test code coverage is provided to ensure a majority of the code has been thoroughly tested.

5.3 Code review

All written code will be reviewed by at least 1 team member before pull requests are accepted. This will ensure the code is comprehensible, working and correct.

Additionally our code will undergo a complete source review by SIG (Software Improvement Group).

5.4 Version control

To maintain a good overview and history of the code we write, we will use a version control system. All our code will be stored on GitHub and therefore use the Git version control system.