Penguard

Distributed Systems – Project Proposal

Nils Leuzinger
14-939-896
nilsl@student.ethz.ch

Nicole Thurnherr
11-925-328
nicoleth@student.ethz.ch

Aline Abler
14-920-979
ablera@student.ethz.ch

Sascha Tribelhorn
13-914-692
tsascha@student.ethz.ch

Luca Wolf
12-913-539
lwolf@student.ethz.ch

ABSTRACT

Don't you know this problem? You're just taking your penguin to party as usual, and all of a sudden it is stolen!

We're tired of this. So now we're going to solve this problem once and for all. Penguard is an application that can prevent such things from happening by alarming us beforehand.

1. INTRODUCTION

Penguard is an application that allows users to monitor objects and informs them when these objects get too far away from them. This can be used as a theft alarm or as an aid to prevent forgetting objects.

While this principle is widely applicable, the idea stems from the need to guard a plush penguin mascot at an event and prevent it from being stolen. In the style of this particular use case, the objects to be monitored will henceforth be called "penguins", while the people monitoring them will be called "guardians".

Penguins can be guarded by several guardians at once, in which case it is sufficient for the penguin to be "seen" by one of them. Also, guardians can monitor several penguins at once. The guardians should know at all times which other guardians see which penguins.

We want to be able to use any kind of bluetooth device as a penguin. The only requirements should be that it supports bluetooth, and that it doesn't automatically turn off bluetooth after a while of inactivity, even when no device is paired to it.

As an extension, penguins should be able to detect when they are "lost" and act upon it. This obviously contradicts the goal to support any bluetooth device. We could support both options by adding an optional protocol, the Penguard Penguin Protocol, through which penguins can communicate with the guardians. When the penguin supports the protocol, it can detect and act upon being lost, otherwise not. We want to keep this option in mind and implement it if we have enough time.

The major challenges will be

- Implementing a peer-to-peer protocol for the guardians
- Detecting bluetooth devices and "tracking" them without making more assumptions on them
- (optional) Implementing a protocol for the penguins to talk to the guardians

2. SYSTEM OVERVIEW

A Penguard group consists of

- One or more guardians guarding the penguins
- Zero or more penguins being guarded

Guardians can be part of at most one group at once. Penguins not supporting the Penguard Penguin Protocol can be part of more than one group–this cannot be prevented. Penguins supporting the Penguard Penguin Protocol can only be part of one group. It is theoretically possible for a second group to add it as a penguin not supporting the protocol, though.

The guardians communicate with each other via Internet using a peer-to-peer protocol (the Penguard Guardian Protocol). They can discover each other by using a Penguard Liaison server. The Liaison server stores the guardians' IP addresses and ports and is used to poke holes when the guardians are behind a NAT.

The guardians can detect whether a penguin is in range by using Bluetooth RSSI. That means that guardians are not required to pair with penguins.

Penguins can optionally implement the Penguard Penguin Protocol, which is a Bluetooth Low Energy protocol. The guardians act as clients to the Penguins. The guardians will ping the Penguins regularly, such that the penguin can detect when it is lost by using a timeout. The penguin should also advertise what kind of information it requires from the guardians, which the guardians then must send to it. This information can include phone numbers, email addresses or similar information and should allow for the penguin to contact the guardians when it is lost.

The guardians should be able to find out whether any given penguin supports the Penguard Penguin Protocol.

2.1 System components

Penguard Android Application is an Android app required to use Penguard. The app can act as a guardian or as a penguin (guardian mode or penguin mode). When acting as a penguin, it optionally supports the Penguard Penguin Protocol. If it does, the user should be able to select what the app should do when the penguin is lost. Options include sounding an alarm and sending GPS coordinates to the guardians via SMS. When acting as a guardian, the application provides an user interface that displays the status of each monitored penguin. The status includes the signal strength (RSSI) and which of the other guardians see that specific penguin.

Penguard Guardian service is part of the Penguard Android Application. It is a service that runs in the background. It handles all Guardian tasks, i.e. monitoring penguins and communicating with other guardians.

Guardians monitor the penguins using the Penguard app in guardian mode. The term "guardian" refers to the Penguard Guardian service or the user using the Penguard Android application

Penguins are bluetooth devices. They are registered with the guardians and then monitored. They optionally

support the Penguard Penguin Protocol, allowing them to take actions when they are lost.

Penguard Liaison Server allows guardians to find each other easily. Guardians register with the Penguard Liaison Server (henceforth referred to as PLS). The server keeps a list of all currently active Penguard users. When starting the Penguard app, guardians will register with the PLS which stores their username, unique UUID and IP address/port. When a guardian wishes to contact another, he asks the PLS to establish a connection. The PLS will look up the other guardian and send him a connection request plus the first guardian's IP/port. The second guardian can then choose to contact the first or not. From that point on, the PLS is no longer involved in communication.

2.2 Calibration (optional)

According to Wikipedia, "There is no standardized relationship of any particular physical parameter to the RSSI reading." [2]. See also [1]. Since the range of RSSI values varies between devices, it should be possible for guardians to calibrate themselves. That is done in a guided process where the penguin is first brought close to the guardian and then slowly carried away. We can then take measurements of the bluetooth signal strength at certain time intervals and calibrate the guardian that way.

2.3 Penguard Guardian Protocol

The Penguard Guardian Protocol (henceforth referred to as PGP) allows guardians to form groups, and it allows guardians within the same group to communicate.

2.3.1 Forming groups

Every guardian is automatically in a group, even when he is alone and not guarding penguins.

Larger groups are formed by merging two existing groups. To do that, one guardian from group A contacts one guardian from group B. The second guardian can confirm or deny the group merge. Once confirmed, the guardian from group B sends all information on group B to the guardian from group A. The latter now knows all the information of the new, merged group, and broadcasts that to every member of the new merged group. All participants then update their status.

2.3.2 Communicating

When a guardian is registered with a group, it will immediately start sending its status to the other group members. The status includes information about which of the penguins currently guarded it sees. It will also receive similar status updates from other group members. These status updates are rather frequent and it won't hurt when some are lost.

Other messages (for example a group merge or addition of a penguin) are more important and it needs to be ensured that every guardian within the group gets them. To ensure this, we plan to implement an atomic commitment protocol. If the transaction aborts, the user initiating it will get an error message and the option to try again.

2.4 Penguard Penguin Protocol (optional)

The Penguard Penguin Protocol (henceforth referred to as PPP) allows for penguins to detect when they are lost.

It is a Bluetooth Low Energy protocol.

The penguin should advertise that he supports the PPP. It can be activated and deactivated.

Once activated, the penguin enters its active state. In this state, it will listen for pings from the guardians and acknowledge them. The guardians must send these pings regularly. When the penguin does not receive a ping for long enough, it will consider itself lost and enter its lost state. Once it receives another ping, it will transition back to active state.

When deactivated, the penguin will go in its inactive state. In that state, the penguin will reply to pings saying that it is inactive.

The penguin can also tell the guardians which information it would like to receive from the guardians. The guardians will poll for this information once. They will then send the required information to the penguin. The required information should not change.

The PPP and all components using it are considered optional, meaning that this will be the first thing we strip of the project when time is not sufficient.

3. REQUIREMENTS

During this project, we will need the following hardware:

- Several Android smartphones
- Several Bluetooth devices
- A server (kindly provided by VSOS)

We will rely on Java for development of the Android app and Python for the Liaison Server.

We will need the following software:

- Android Studio

4. WORK PACKAGES

- **WP1**: Local penguin tracking
- **WP2**: Low-level networking
- **WP3**: PGP - protocol design
- **WP4**: Penguard guardian service implementation
- **WP5**: PLS implementation
- **WP6**: Functional graphical user interface for the Penguard app
- **WP7**: (optional) calibration functionality for guardians
- **WP8**: (optional) PPP - protocol design and implementation
- **WP9**: Field testing
- **WP10**: Presentation

5. DISTRIBUTION OF WORK

In this section we will list the distribution of work packages on our group members. This distribution is not definitive, but should serve as a guideline for who works on which parts, and, most importantly, how many people will need to work on the individual work packages. Generally, we have assigned people according to their interests. We have tried to give everyone an equal share of work. The optional work packages are not listed, since those will be used to even out the amount of work everyone has.

- **WP1**: Nicole
- **WP2**: Nils
- **WP3**: Aline, Sascha, Luca
- **WP4**: Aline, Sascha
- **WP5**: Luca
- **WP6**: Nicole, Nils
- **WP9**: Everyone
- **WP10**: Everyone

6. MILESTONES

- **Phase 1**: Define goals and work plan
- **Phase 2**: WP1, WP2, WP3–Protocol design (PGP), Intent flow design (Android app), implement penguin scan and discovery routines, implement simple networking routines in Android app (packet dispatcher and listener).
- **Phase 3**: WP4, WP5–Implement Penguard Guardian service (routines for sending status messages, atomic commitment protocol), implement PLS
- **Phase 4**: WP4, WP6, WP7–Implement various events (group merge, penguin add...), refine UI, implement calibration routine
- **Phase 5** (optional): WP8–Design and implement PPP
- **Phase 6**: WP9, WP10–Extensive field testing, prepare presentation

7. REFERENCES

[1] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos. Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization, 2011.

[2] Wikipedia. Received signal strength indication, 2016.

8. DELIVERIES

We expect to deliver the following:

- Code for the Penguard Android application
- Code for the PLS
- Documentation for the PGP
- (optional) Documentation for the PPP
- Slides for the one-minute presentation