Mini-project proposals 2021

Rune Hylsberg Jacobsen

Version 0.9 – September 13, 2021

Introduction

Mini-projects are an essential part of the Internet of Things (IoT) Technology course. The mini-projects offer an opportunity to get hands-on experience with some of the technologies taught in the course. The projects are targeting student groups of 3-5 members. In the first part of the course, groups are formed and will select a mini-project that they should work with during the course.

There are a few general criteria that you should note when defining, conducting, and reporting from the mini-projects.

- **Problem-driven** – Mini-projects should be driven by a clear and challenging problem relating to the industry or society at large. Therefore describe and motivate for the problem that you want to investigate.
- **Relevance** – It should be made clear how a mini-project links to one or more key topics from the course.
- **Methodology** – The mini-project should demonstrate a scientific practice. This includes a (brief) introduction to the scientific knowledge base on which the project rests as well as en emphasise on the advancement of this knowledge. Furthermore, research questions/hypotheses, metrics and data for experiments should be presented.
- **Analysis** – The mini-project should provide analysis of concepts, ideas, a design or a solution, experimental results, and/or contribute with a comparison to similar work.
- **Dissemination** – Make the written presentation of your project concise. You should only briefly describe the background of your topics and instead focus on the results and the parts you made in the project.
- **Individual contributions** – For each project member, add a short statement describing his/her main contributions to the project.

Project 1: LoRa channel characterization

Maximum number of groups enrolling in this project: 1

Prerequisite: General wireless communication technology

This project takes a deeper look at the LoRa technology. The objective of the project is assessing the suitability of LoRa radio technology for IoT applications. The project aims to characterize the radio channel for LoRa operating at 868 MHz. Focus for the characterization is signal-strength, path loss, delay, delay variation, and throughput. Measurements are done both in open environments and in addition also in shallow layers of soil. The attenuation of the radio signal in the soil is calculated.

The project involves hands-on prototyping with a Raspberry Pi computer platform extended with a LoRa hat operating at 868 MHz [1, 2]. (Fig. 1). It includes development of test and data

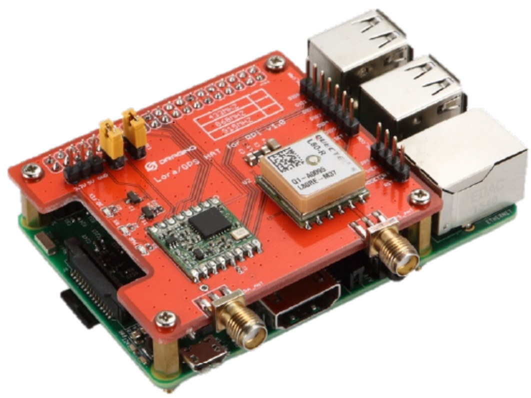

Figure 1: LoRa and GPS hat module stacked on a Raspberry Pi 3.

acquisition software, and descriptive statistical analysis. For distance measurements in the open field, the build-in GPS receiver can be used.

Pivot points of the project

Try to address as many of the following pivot points of the project.

- Create a small testbed with a LoRa radio transmitter and receiver.
- Verify the Chirp Spread Spectrum (CSS) of the LoRa communications with an oscilloscope (optional).
- Perform measurements of Received Signal Strength Indicator (RSSI), delay and delay variations for different distances between transmitter and receiver and with different spreading factors.
- Make statistical analysis of the measurements.
- Calculate the path loss for the channel and estimate the path loss exponent for each environment used for testing.
- Organize and document your code for running your LoRa testbed in a software repository such as Git.

Reporting

The report should be written in a self-contained style addressing the general criteria listed in the introduction section. In your reporting, your reporting should address the following:

- Summarize the theory of LoRa and present the formulas to calculate throughput and Airtime of LoRa for varying values of the spreading factor.
- Discuss IoT applications that may benefit from a wireless communication using LoRa.

References

[1] CNX Software, Dragino LoRa/GPS HAT Board for Raspberry Pi, `https://www.cnx-software.com/2016/07/27/dragino-loragps-hat-board-for-raspberry-pi-sells-for-32/`, accessed 2021-10-12

[2] Dragino Wiki page, `https://wiki.dragino.com/index.php?title=Lora/GPS_HAT`, accessed 2021-08-12

Project 2. Routing simulator

Maximum number of groups enrolling in this project: no limit

Prerequisite: Basic Python and object-oriented programming skills

SimPy is process-based discrete event simulation framework for Python [1]. It has the potential of simulating protocol dynamics of IoT networks.

This project designs and demonstrates a routing simulation using PySim. The aim is to evaluate the Routing over Low Power Lossy Networks (RPL) protocol [2]. The project may start with a simplified version of the RPL demonstrating the establishment of the DODAG i.e, the DIS and the DIO message (Fig. 2). Hereafter, the progress with the distribution of network prefixes using DAO messages and extending the DIO messages with the trickle algorithm [3].

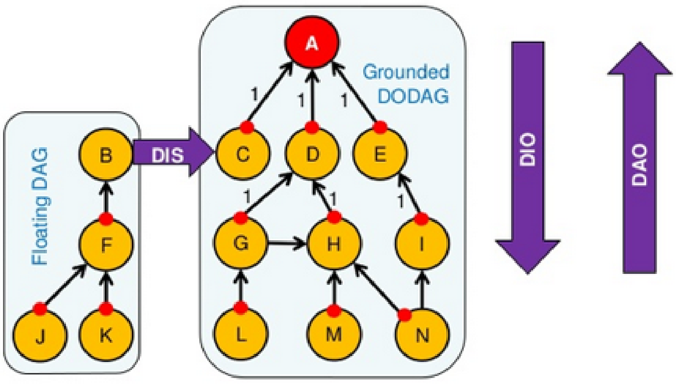

Figure 2: Example of the RPL routing protocol and its use of DIS and DIO messages to build and maintain the DODAG.

Pivot points of the project

Try to address as many of the following pivot points of the project.

- Establish a network and a node concept in SimPy.
- Implement and simulate a neighbor discovery mechanism that ensures that each node establish connectivity with its nearest neighbors.
- Implement the DIO message and simulate the DODAG formation with a varying number of nodes with different sets of neighbor nodes.
- Implement the DIS message (optional).
- Implement a trickle algorithm [3] to control the period of DIS messages in the network (optional).
- Design a node addressing plan for the network. Make assumption on the use of a mesh under or a route over architecture.
- Implement the DAO message and demonstrate the distribution of network prefixes in the DODAG.
- Simulate a DODAG local repair mechanism (optional).
- Organize and document your code for the simulator in a software repository such as Git.

Reporting

The report should be written in a self-contained style addressing the general criteria listed in the introduction section. In your reporting, your reporting should address the following:

- A brief introduction to RPL.
- Clarify of the performance metrics used for the protocol evaluation.
- A description of the implementation.
- Simulation results and their analysis interpretation.

References

[1] SimPy, Discrete event simulation for Python, `https://simpy.readthedocs.io/`, accessed 2021-08-12.

[2] T. Winter, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, Internet Society, RFC 6550, March 2012.

[3] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, The Trickle Algorithm, Internet Sociefy, RFC 6206, March 2011.

Project 3: 6LoWPAN networking

Maximum number of groups enrolling in this project: 2

Prerequisite: IPv6 networking and hands-on experience with Linux networking

The project aims to demonstrate and evaluate 6LoWPAN on the Raspberry Pi (RPI) with one RPI acting as 6LoWPAN gateway tunneling packets to the IPv4 internet. The project uses IEEE 802.15.4 radios from openlabs.co that can be added to the RPI hardware [1]. Besides connectivity the project should aim to explore demonstrate a number of aspects of the 6LoWPAN adaption layer including fragmentation.

Figure 3: IEEE 802.15.4-radio tranceiver from Openlabs.

Pivot points of the project

Try to address as many of the following pivot points of the project.

- Setup of a 6LoWPAN network consisteing of at least 3 nodes.
- Use a packet analyzer e.g., WireShark to analyze the IPv6 and the 6LoWPAN packets in the network (requires additional hardware TBD).
- Use a network testing tool such as iperf [2] to characterize traffic in the networks.
- Compare network performance for IEEE 802.15.4 running in beacon and beaconless mode (optional).
- Implement a NAT64 gateway to provide interoperability between IPv4 and IPv6 (optional). The NAT64 can be implemented on the border router using the Tayga software.
- Organize and document your code for running the 6LoWPAN testbed in a software repository such as Git.

Reporting

The report should be written in a self-contained style addressing the general criteria listed in the introduction section. In your reporting, your reporting should address the following:

- Brief introduction to 6LoWPAN and the IEEE 802.15.4 radio technology.
- Clarify of the performance metrics used for the protocol evaluation.
- Description of the implementation.
- Presentation and discussion of results from network tests.

References

[1] RAspberry PI 802.15.4 radio, openlabs.co, http://openlabs.co/OSHW/Raspberry-Pi-802.15.4-radio, accessed 2021-08-12.

[2] iPerf, https://iperf.fr/, accessed 2021-08-12.

Project 4: Web-of-things

Maximum number of groups enrolling in this project: 4

Prerequisite: Basic programming and software handling skills

The aim of this project is to create a simple RESTful IoT application based on a sensor devices (smart object) providing continuous time-series data reports. The project uses a Message-oriented Middleware (MoM) to handle multiple time-series data streams.

For the sensor devices two variants exits with different hardware configuration for the IoT devices (Fig.4):

- **Variant 1:** Weather station based on the SenseHat to the Raspberry Pi [1]. This project variant may use inspiration from [2, Sec. 9.4]
- **Variant 2:** Onboard Diagnostics (OBD) interface [3]. This variant of the project requires that you have access to car that has the OBD II interface.

a)

b)

c)

Figure 4: a) SenseHat for RPI for the weather station. b) A PiCAN 2 HAT provides the RPI with full CAN-Bus capability. c) Photo of OBD physical interface in a car.

Pivot points of the project

Try to address as many of the following pivot points of the project.

- Create an IoT device and a setup that can report continuous time-series data.
- Establish a MoM e.q., based on MQTT [5].
- Demonstrate time-series data flow for subscribers of different flows.
- Design a data model for your IoT data.
- Make simple analytic functions acting on time-series data in near real-time.

- Organize and document your code for running your IoT application in a software repository such as Git.

Reporting

The report should be written in a self-contained style addressing the general criteria listed in the introduction section. In your reporting, your reporting should address the following:

- Variant 1: Explain the CAN protocol briefly.
- Variant 2: Review briefly IoT application that combines time-series data and geolocation.
- Discuss scalability of your system.
- Document the REST API interface of your system.

References

[1] https://www.raspberrypi.org/products/sense-hat/, accessed 2021-08-12.

[2] Arshdeep Bahga, Vijay Madisetti, Internet of Things: A Hands-On Approach, 2014, http://www.hands-on-books-series.com/iot.html

[3] Elector Mag, PiCAN 2 – CAN-Bus Board for Raspberry Pi (OBDII), https://www.elektormagazine.com/news/pican-2-can-bus-board-for-raspberry-pi, accessed 2021-08-12.

[4] https://www.iso.org/standard/67245.html

[5] https://www.iso.org/standard/69466.html

Project 5: Asymmetric encryption on a constrained device

Maximum number of groups enrolling in this project: no limit

Prerequisite: Preferable C/C++ programming skills. Alternative Python programming skills.

RSA and ECC can be used as a public-key encryption mechanism for constrained devices. ECC is attractive due to its comparative shorter keys. The project provides an overview of crypto-libraries that support RSA and ECC and can be implemented on the Arduino Due, RPI, or similar. It furthermore demonstrates a possible implementation and provides a selected set of performance measurements.

The preferred implementation is in C/C++. The RELIC library [1] may be a good starting point for such implementation. As an alternative Python may be used with for instance with the PyCrypto [2] and the PyECC library [3]. For both C/C++ and Python implementation you may find other suitable libraries.

Pivot points of the project

Try to address as many of the following pivot points of the project.

- Run an RSA and an ECC cryptosystem on one or more constrained device targets.

- Perform timing measurements for encrypt, decrypt and key generation algorithms. Vary the size of payload data and key sizes.
- Make an interoperability test with another group (optional).
- Compare the performance of a number of different library implementation (optional).
- Search and implement other asymmetric ciphers besides from RSA and ECC (optional). An example is Elgamal.
- Organize and document your code in a software repository such as Git.

Reporting

In your reporting, your reporting should address the following:

- Brief introduction of relevant asymmetric ciphers.
- Description of the implementation including details on how reliable an accurate timing measurements has been implemented.
- Presentation of results from performance measurements asymmetric ciphers including comparison of different library implementations if applicable.

References

[1] D. F. Aranha and C. P. L. Gouvea, RELIC is an Effcient LIbrary for Cryptography, 2020, `https://github.com/relic-toolkit/relic`, accessed 2021-08-12.

[2] Darsey Litzenberger, PyCrypto - The Python Cryptography Toolkit, 2014, `https://www.dlitz.net/software/pycrypto/`, accessed 2021-08-12

[3] R. Tyler Ballance and B. Poettering. PyECC: Python Elliptical Curve Cryptography, 2009, `https://github.com/slideinc/PyECC`. Accessed 2021-08-12

Report guidelines and recommendations

This section gives a set of guidelines for the mini-project report. It is a general guideline and that some projects may have (good) reasons for deviating from these guidelines.

Reports should be written in single column with a 12 pt font size and approximately 2,5 cm of margin (left, right, top and bottom). There should be no page break between sections. the length of the report should be about 8 pages including figures, tables and references. Do not use page breaks before a new sections.

When writing the report you may assume that the reader has a background in networks, protocols and systems corresponding to having followed the course. The reader cannot be assumed to an expert in the particular technology behind your project.

The report should be written in English and should be organized into the following sections:

- **Title**. Print the title of you project and clearly list the names of the project participants as well as the date. This is not a page on its own.

- **Introduction** [approximately 1 page]. This section must give a brief motivation and introduction to the topic of the project. The section should also give a brief account of the results that have been obtained. Emphasize on would course element you are addressing and how you address them.
- **Background** [approximately 1 page]. This section should contain the technical background of the project required to understand the rest of the report. It will typically explain the protocols being implemented, and the software/hardware/technology being used. Note that if you have many interesting results, this is the section to slim or optionally leave out.
- **Design, implementation , and test setup** [approximately 2 pages]. This section must explain the design and implementation of the "product/service" that you are creating in the project. It must present the main design choices and assumptions that you have made. It must give an overview of the implementation made and explain in detail the key components of your implementation/test setup. When submitting the your report please also provide a link to where the source code etc. for the project can be found.
- **Experiments** [approximately 3 pages]. This section must document the experiments that you have made and the results that was obtained. It should also give a motivation for the experiments that have been made. Use plots and tables to present your results.
- **Conclusion and future work** [approximately 1 page] This section should sum up the work that has been done and the main results that have been obtained. It should also give some pointers towards future work in the area.
- **References**. You should provide a set of key references for the project. This section is not included in the page count.
- **Appendicies**. This may or may not be needed. Appendices are not included in the page count.