Danmarks Tekniske Universitet

Report 1

Course:
02450: Machine Learning

Author(s):

Anders H. Opstrup (s160148)
Gu Jinshan (s161944)
Huayu Zheng(s126077)

DTU Compute
Institut for Matematik og Computer Science

02450: Machine Learning
Tue Herlau

27 September 2016

Contents

1 Introduction

In 2010 Cubesat Space Protocol was created at Aalborg University. The protocol was created to use with the CubeSat satellite form in general and not only for Aalborg University. Cubesat Space Protocol has support for mechanisms such as packet checksums build in, and amongst other supports encryption. To make the protocol as generic as possible, the protocol was created with support for the operating systems Linux and FreeRTOS (later Mac OS X and Windows). Cubesat Space Protocol is designed with concurrency in mind, using the thread libraries of the relevant operating systems, and as of now no single threaded version of the protocol exists.

A CubeSat is a small research satellite contained in a 10cm ×10cm ×10cm box using mostly off the shelf electronic parts.

At DTU eCos is used as operating system on satellites. eCos is a real time openSource operating system, mainly intended for embedded use. eCos is highly customizable and can thus be designed for at very low resource footprint, making its use on satellites among other things a viable solution. eCos is based on GNU tools and bases its nature around having a POSIX compatible API. One of the key features in this situation is its support for threads, which is needed for implementing Cubesat Space Protocol which this project does.

This report will firstly outline some needed background knowledge in section **??**. Section **??** describes the outline of the design requirements. One of the original reasons for this project, was to have Error Correcting Codes for the satellite link, but the goal has moved towards documenting the current protocol used at DTU and comparing it to Cubesat Space Protocol. This is done in section **??**. Implementation specific details can be found in section **??**, while details on testing the implementation comes in section **??**. Section **??** describes the changing goal of the project in more detail along with ideas for ECC, and section **??** concludes the project.

This document requires previous knowledge to the protocols TCP and UDP and also the OSI reference model. Basic ECC understanding is needed to understand section **??**.

2 Conclusion

While looking to implement Cubesat Space Protocol on eCos, it turned out that the current protocol used at DTU, needed better documentation in order to compare it to Cubesat Space Protocol. Before documenting and comparing these protocols, a brief introduction to RDP was given with comparison with TCP and UDP. After creating the documentation for dtuProto it was compared to Cubesat Space Protocol, but no protocol has been chosen as the "correct to use" protocol, as it is not the scope of this project and seems like a question of satellite design rather than best designed protocol.

Implementing Cubesat Space Protocol on eCos has been successful, and running test programs has shown the same behaviour as a Linux version of Cubesat Space Protocol indicating a working implementation. Some extra parts of Cubesat Space Protocol has not been tested as no test programs for these exist and some compiler issues prevents this. These should be tested thoroughly when these issues has been fixed, but operation of Cubesat Space Protocol with CSP-UDP is possible.

A note on how ECC could be used alongside dtuProto and Cubesat Space Protocol has been given. The use of the FX.25 opposed to other ECC algorithms has been touched. The use of an in-radio-interface implementation seems to be the best option when it comes to ECC.