Contribution ID: **1**   Type: **Talk**
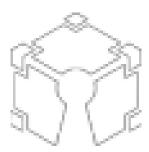
An Open Source Implementation of the CCSDS SLE Services in Python

Monday, 14 October 2019 09:30 (20 minutes)

An Open Source Implementation of the CCSDS SLE Services in Python

Milenko Starcik (1), Fabian Burger (1), Artur Scholz (1, 2), Tiago Nogueira (1)
(1) VisionSpace Technologies GmbH
(2) LibreCube Initiative

The Open Source CubeSat community is currently lacking the capability to use existing radio amateur ground-station networks to transfer live data to the end users / mission control centres. SatNOGS, the biggest network of the kind, provides services for archiving and offline distribution of telemetry data. At the moment, an operator using the SatNOGS network is not able to receive telemetry in real-time and make use of the station uplink chain, when available, for direct commanding. We believe that the community could benefit considerably from services that provide real-time monitoring and control over a distributed ground-station network.

As a first step to address the current limitations we introduce in this contribution a new free and open source implementation of the CCSDS Space Link Extension (SLE) services in Python. The SLE standard, used by all major space agencies, is one of the most adopted CCSDS standards and is key to enable the inter-agency utilisation of ground-station networks like the Deep Space Network. Additionally, it is supported by most, if not all, private ground-station operators. SLE services have proven reliable, and can be used even for missions, like most CubeSats, that do not adhere to the CCSDS telemetry and telecommand frame and packet formats. SLE provides different types of telemetry return services, for real-time telemetry reception, a telecommand forward service and the capability to receive annotation information about the current status of the space link. We take the initiative to put forward a solution based on SLE because of its proven track record and out-of-the-box interoperability with existing commercial and agency ground-stations.

Our current SLE stack offers:
1. An SLE provider, offering the ground-station (provider) side services for delivery of telemetry and uplink of telecommands. In addition, it offers some level of ground-station configuration management services over a REST API (https://github.com/visionspacetec/sle-provider)
2. An SLE management client, allowing anyone to access the SLE management services on the provider using a Python API (https://github.com/visionspacetec/sle-management-client)
3. A set of SLE common libraries (e.g. ASN.1 mapping) for the development of provider and user services (https://github.com/visionspacetec/sle-common)

In the current implementation only the Return All Frames (RAF) service has been implemented. RAF is not only able to transfer CCSDS telemetry transfer frames but also any other frame format like AX.25, which is used by many amateur CubeSats. The event-loop based implementation in Python is lightweight, cross-platform and easy to deploy. Python was chosen because of its wide use in the Open Source community and the short development cycles it allows. It provides high data throughput, enough for amateur and most commercial satellites. A heartbeat mechanism ensures that the connection is always alive and ready to transfer mission critical data. The communication between user and provider can be encrypted, using the implemented protocol features and has been verified with up to date reference software from the European Space Agency (ESA). For the REST API, the use of HTTPS is prepared and can always be combined with an authentication scheme to ensure data integrity at any time.

The current stack does not offer an implementation of an SLE user since at the time of writing at least two free and open source implementations exist: LibreCube SLE and the NASA AMMOS framework. Our provider has been tested and proven to work with both. In addition, we have tested our implementation against the SCOS-2000 ground-station Network Interface System (NIS), showing that it holds against existing commercial implementations of the SLE user. In what concerns the interface of the SLE provider with the ground-station backend, the provider has been tested with the Cortex CRT at ESA's European Space Operations Centre (ESOC) development ground station ESOC-1.

Besides its use in a traditional ground-station setup, the SLE provider can also be used to relay telemetry and telecommand data from/to cloud ground-station service providers and satellite operators. Cloud based scheduling systems could be used to dynamically configure the ground-station network, using the SLE provider's REST API, ensuring highly efficient use of the ground-station time and minimizing the need for human interaction. Upcoming developments include not only the support for a wide range of commercial and amateur ground-station equipment and Software Defined Radios (SDR), but also extensive testing and further performance improvements. Implementation of the telecommand forwarding service is also planned.

Primary authors: STARCIK, Milenko (VisionSpace Technologies); Mr BURGER, Fabian (VisionSpace Technologies); Mr SCHOLZ, Artur (LibreCube Initiative); Mr NOGUEIRA, Tiago (VisionSpace Technologies)

Presenter: STARCIK, Milenko (VisionSpace Technologies)

Session Classification: Talks

Track Classification: Software