GROUND STATIONS NETWORK – A VIABLE SOLUTION

Alexandru BARBOVSCHI, astronomical observatory coordinator; Andrei MARGARINT, ground stations coordinator

Technical University of Moldova

Abstract: *Each satellite needs at least one possibility of communication with Earth to deliver the data obtained in space. So, at least one ground station is necessary per satellite. Further ground stations extend the communication duration with a satellite if placed at the correct locations on Earth. The idle times of a ground station between passes and the costs for multiple ground stations per mission can be decreased if several ground stations can be shared between several satellite missions. This idea leads to ground station networks. Otherwise, if several ground stations are available in different locations in a ground station network, the data received from several of them can be combined to increase the signal quality and to yield less outage.*

Key words: *satellite, network, ground, station, education, server, client, software*

1. Introduction

CubeSats and nanosatellites in a Low Earth Orbit offer a cheaper alternative and feature faster development cycles than traditional space missions and therefore give access to space science for education and research at universities. For students, satellite missions represent a challenging exercise which is ideal for problem based learning. Satellites require lots of interdisciplinary skills, teamwork and project management. The satellite's design, building and testing can be broken down into subprojects which can then be solved by different teams with different states of knowledge or with different locations. With a few exceptions, all CubeSats have been built in the context of universities.

Not only satellites are well suited for the training of students – ground stations offer an even better learning basis for students in telecommunications because they are easier accessible than satellites. This is facilitated by the modular structure of the ground station and by the open interface approach between the components and especially between the segments. Students can experience a live, real world satellite link, can set up a separate PC in parallel to the actual mission PCs or data recordings of satellite passes can be used in a playback scenario by students to simulate a satellite. By this, the operation of a ground station can be learned.

Satellites need at least one possibility of communication with ground to deliver the data obtained in space. So, at least one ground station is necessary per satellite. Further ground stations extend the communication coverage with satellites if placed at the correct locations over the planet. The idle times of a ground station between passes and the costs for multiple ground stations per mission can be decreased if several ground stations can be shared between several satellite missions. This idea leads to ground stations connected into a network.

2. Solution proposal

In order to implement such a network of ground stations an architecture must be chosen. It was decided that a classical server-client architecture will fit the needs of the project. At the global scale there are three parts of the network – a VPN server, a separate network device which inter-connects main server and clients in a secure manner; main server, a high-performance PC which manages the entire network and gives a web interface for the end-users; clients, a series of PCs around the world, connected to the VPN network. Clients can be of two types: just for accessing the web interface and/or connecting a ground station to the network for shared usage.

3. Current architecture implementation

In the previous part there were enumerated three parts of the system. Detailed description follows onward.

As it was mentioned a VPN server is a separated network device. And it is indeed at the current stage of the implementation – VPN server runs on a MikroTik Cloud Router. With amazing performance, high level of flexibility and big range of possibilities it perfectly fits the project's needs.

Main server runs on a Sun Microsystems blade server. Ubuntu Server LTS is used as operating system and is running a series of custom services developed for the operation of the network. There is a second identical blade server for redundancy purpose.

Client machine can be any kind of PC, ranging from a low-performance SBC to a high-end desktop computer. The choice depends on the end-user's purpose – will the machine be used only for accessing the web interface and/or connecting the ground station to the network. At the current stage of implementation client software for connecting ground station to the network was successfully tested on a Raspberry Pi SBC and a usual desktop PC, both running an Ubuntu distribution. Theoretically it can run on any major operating system, which includes Microsoft Windows, Mac OS X, GNU/Linux and even on *BSD derivatives. Testing of this claim is a part of future development cycles.

4. Main server description

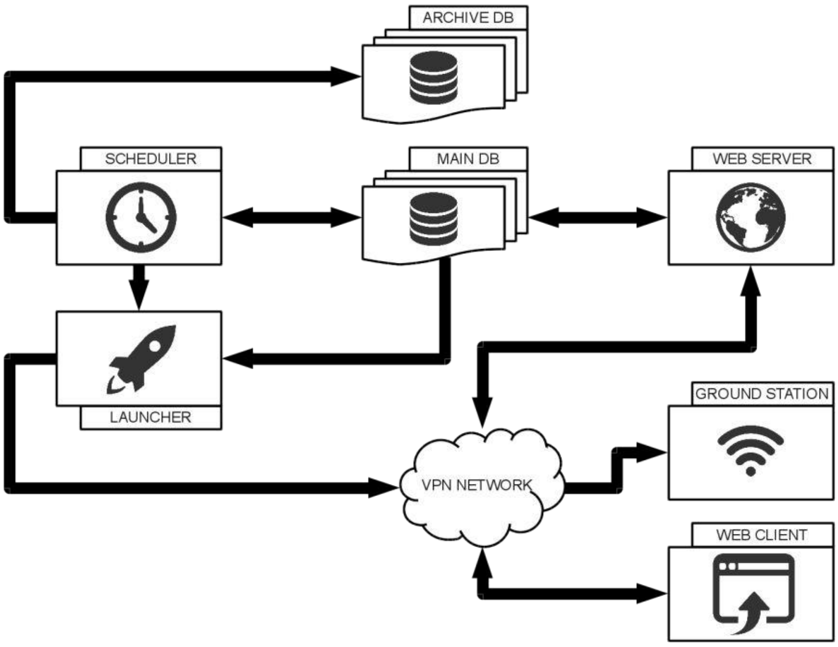

Figure 1 Overview of the server-side software

Server-side software represents a series of services:

– Main DB. Its role is storing the actual data necessary for serving current operations (the stored data is not older than of 2 weeks);

– Scheduler. This component is accomplishing several tasks. First one is monitoring the Main DB for the fresh data. The second is taking decision based on previously acquired data – calling the Launcher or sending the data to Archive DB. Third one is keeping updated all the TLEs and future observations based on updated information;

– Archive DB. In this database is stored data older than 2 weeks, in order to preserve high performance of the Main DB;

– Web Client. This is the GUI part of the system which allows end-users to interact with the system – schedule new observations, view or remove the old ones, view information about connected ground stations etc. It can be accessed only by the clients of the VPN Network;

– Web Server. Web Client is served by this component, giving it all the functionality and access to the databases;

– Launcher. This service is the one that actually communicates with clients (those with ground station connected), sending them commands necessary for accomplishing a specific task, based on parameters given by Scheduler;

– Ground Station is the end-point of the system which receives the commands sent and executes them.

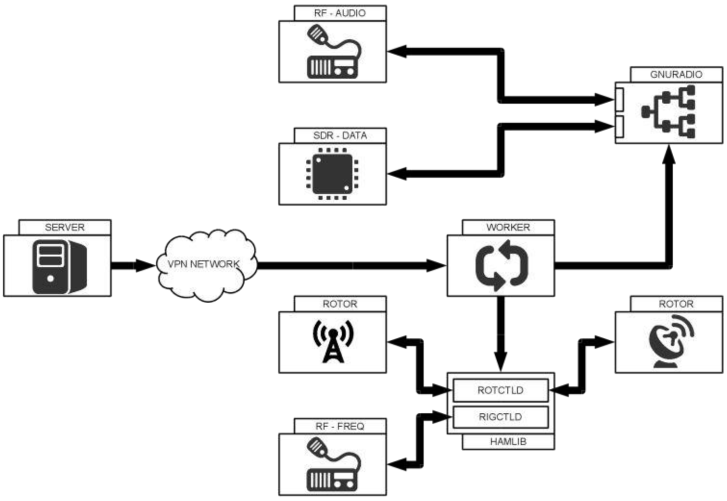

Figure 2 Overview of the client-side software

5. Client-side software

Client-side is a bit more complex than server-side. It includes not only PC but also the hardware connected to it (rotor, transceiver). Components of the client-side are as follow:

– Worker. This service is the main one and it controls all the others. Its role is sending commands to several daemons;

– Hamlib. This open-source project contains 2 sub-components – rotctld and rigctld. Rotctld is responsible for communicating and controlling rotors of different types. Rigctld is able to control many types of transceivers, giving the ability, for example, to change their working frequency;

– Rotor. As mentioned, they are controlled by rotctld that is able to control different types of rotors, including the ones for telemetry antennas with low angular precision and also the ones with high angular precision for parabolic antennas;

– RF – Freq. As it was mentioned above, rigctld is able to control working frequency of various transceivers;

– GNU Radio. This open-source project has very wide abilities of communicating with various types of telecommunication hardware, giving the possibility for receiving of signals with advanced post-processing and also sending of signals with desired pre-processing. It can be used for receiving and processing signals in form of an audio stream from transceivers or raw data from, for example, SDR devices;

– RF – Audio. As mentioned, GNU Radio is taking care of receiving and processing signals received by the means of an audio channel from transceiver;

– SDR – Data. GNU Radio is processing raw data coming from a connected SDR device. This SDR device can, for example, receive the data on 5.8GHz band, which represents a high-speed communication channel;

– VPN Network. Client-server communications are completely tunneled through secure VPN network;

– Server. Client receives all of the commands from server-side. But in case of communication interruption, client will be able to finish its current task.

6. Current software implementation and future plans

For the server-side software all components were implemented, except the part related to Archive DB. As for the client-side all the components were implemented. Important note – GNU Radio component is extensible by any end-user, in order to achieve desired task. This is one of the main educational components.

Future plans includes implementing the Archive DB part, improving and extending hardware support and, based on external partners' feedback, enhancing the Web Client functionality for better user experience.

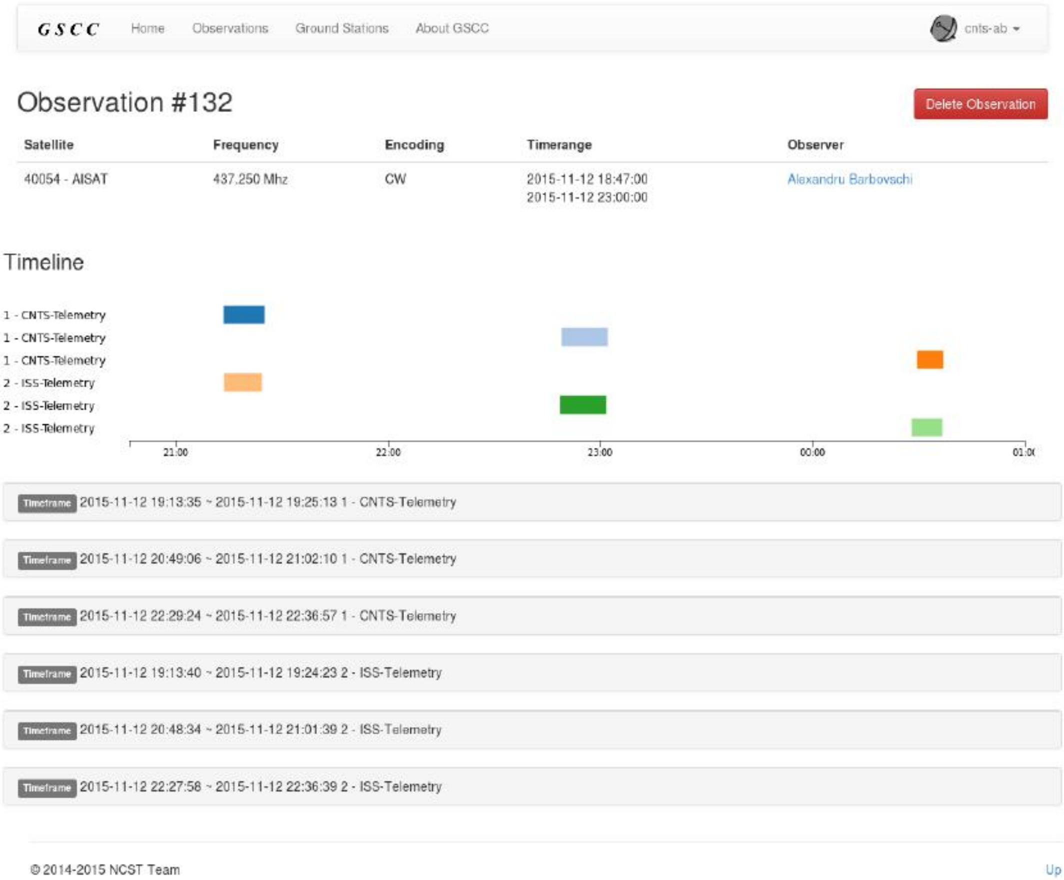

Figure 3 An example of scheduled observation, made using web interface provided by server-side

Bibliography

1. T. S. Tuli, N. G. Orr, and R. E. Zee, "Low Cost Ground Station Design for Nanosatellite Missions" in *Proc. of 24th AMSAT North American Space Symposium*, San Francisco, CA, USA, October 5–11, 2006.
2. M. Kasal, "Experimental Satellite Laboratory BUT Education and Research," in *Proc. of 17th International Conference on Microwaves, Radar and Wireless Communications (MIKON 2008)*, pp. 367–374, Warsaw, Poland, May 19–21, 2008.
3. B. Klofas, "A Survey of CubeSat Communication Systems," in *5th CubeSat Developers Conference*, Cal Poly, San Luis Obispo, CA, USA, April 9–11, 2008.
4. J. W. Cutler and C. A. Kitts, "Mercury: A Satellite Ground Station Control System," in *Proceedings of the IEEE Aerospace Conference 1999*, pp. 51–58, Snowmass at Aspen, Colorado, USA, March 6–13, 1999.
5. Y. Peng, *Ground Station Design for UWE-1 (University of Wüzburg's Experimental-Satellite 1)* Master's thesis, Luleå University of Technology, Department of Space Science, Luleå/Kiruna, Sweden, 2007.
6. Official web site of the Ham Radio Control Libraries: http://sourceforge.net/projects/hamlib/
7. Official web site of the GNU Radio: http://gnuradio.org/redmine/projects/gnuradio/wiki